The Dissertation Committee for Martin Cornelius Heyne
certifies that this is the approved version of the following dissertation:

Spacecraft Precision Entry Navigation using an Adaptive Sigma Point Kalman Filter Bank

Committee:

Robert H. Bishop, Supervisor

David G. Hull

Glenn Lightsey

Todd A. Ely

Glenn E. Peterson

Spacecraft Precision Entry Navigation using an Adaptive Sigma Point Kalman Filter Bank

by

Martin Cornelius Heyne, Dipl.-Ing.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN
May 2007

. . . and behold, a throne stood in heaven, with one seated on the throne. . . . and around the throne was a rainbow that had the appearance of an emerald. Around the throne were twenty-four thrones, and seated on the thrones were twenty-four elders, . . . the twenty-four elders fall down before him who is seated on the throne and worship him who lives forever and ever. They cast their crowns before the throne, saying,

"Worthy are you, our Lord and God,
to receive glory and honor and power,
for you created all things,
and by your will they existed and were created."

-*Revelation* 4

Acknowledgments

I wish to thank the multitudes of people who helped me. I would especially like to thank my advisor, Dr. Robert H. Bishop, for his generous time and commitment. Throughout my doctoral work he encouraged me to develop my ideas research skills. He continually stimulated my analytical thinking and greatly assisted me with scientific writing. I am also very grateful for my doctoral committee and wish to thank Dr. David G. Hull, Dr. Glenn Lightsey, Dr. Todd A. Ely and Dr. Glenn E. Peterson for their continued support and encouragement.

I wish to thank James D. Hillhouse IV for assisting me on many occasions regarding my computational needs to perform my work. I would like to thank all my colleagues and supporters who have advised and supported me though the process of completing the dissertation. They were of great help for many portions of my research work.

Also, I would like to thank my family. My father for continuously supporting my passion for space exploration. My mother for her support despite the great distance separating us and her commitments to my sister and nephew. Thanks to my sister for her continuous kindness and encouragement keeping my life in proper perspective. Finally thanks to Andres and Sarah for being there to keep my life in balance during this effort.

This work was supported by the NASA Mars Technology Program, Advanced Entry, Descent, and Landing Task, California Institute of Technology Grant No. 1263403.

Spacecraft Precision Entry Navigation using an Adaptive Sigma Point Kalman Filter Bank

Publication No. ____________________

Martin Cornelius Heyne, Ph.D.
The University of Texas at Austin, 2007

Supervisor: Robert H. Bishop

This work documents the development of a sigma point Kalman filter for the purpose of precision spacecraft navigation during the atmospheric entry, descent and landing phase. The use of the sigma point Kalman filter is driven by the desire to avoid complex partial derivatives associated with the standard extended Kalman filter. The strategy increases the likelihood that the navigation algorithm will be compatible with the Electra.

Using Mars Exploration Rover Spirit (MER-A) and the Mars Science Laboratory (MSL) data, experiments were conducted to validate the proposed navigation concept. Beginning at atmospheric entry interface, the hypersonic entry phase is considered and the navigation architecture performance is quantified.

Using the sigma point Kalman filter as the main computational unit, a filter bank for environmental parameter identification is investigated. The focus of the investigation is atmospheric parameter identification. The MER-A mission is used to verify the ability of the filter bank to make appropriate selections.

The navigation architecture is implemented on the Electra programmable radio, a flight hardware communication node available on spacecraft build for Mars exploration. The investigations show that the sigma point Kalman filter structure is very applicable to the atmospheric entry navigation problem. When used in conjunction with the filter bank concept, the overall navigation architecture is shown to be able to improve navigation accuracy over standard dead-reckoning, while providing robustness to uncertainties in the atmosphere. The navigation algorithm is successfully hosted on the Electra programmable radio and is capable of processing actual MER inertial measurement data.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

With ever increasing demands being levied on spacecraft entry, descent and landing (EDL) systems, it is essential to the success of many missions that the spacecraft be capable of precisely arriving at a site on the surface of a planet. In light of the vast distances that can separate a spacecraft from the Earth-based control center and the speed at which decisions must be made during the short time of the EDL phase [11], [12], it is necessary that the spacecraft operate autonomously. In the case of planetary exploration where the atmosphere is uncertain and highly variable, the issue of navigation robustness in the presence of uncertainties must be addressed. The development of a navigation architecture is presented here that utilizes advanced navigation architectures [10] which are broadly applicable to most planetary EDL problems where autonomy is desired and flexibility in regard to model uncertainties and vehicle as well as environmental mis-specifications are encountered.

1.1 Entry, Descent and Landing

Future planetary exploration missions, especially to the planet Mars will demand the ability to reach specific regions on the surface of the planet.

This is driven by the fact that science will have questions regarding particular localized regions as high quality surface maps of Mars have pointed to regions of geological interest. Furthermore, potential surface rendezvous requirements may demand a spacecraft land near a rover, drilling platform, or outpost on the surface. These potential interests will require precision landing capability for the spacecraft.

Current navigation methods are not able to meet these precision requirements [23]. For distant planets such as Mars, autonomy and adaptability is essential for the navigation architecture as signal travel times are too great for control from earth. The spacecraft has to be able to adjust to the actual environment encountered and achieve the required precision constraints autonomously [9]. The development of highly precise systems that function without operator input for GN&C problems associated with autonomous interplanetary spacecraft require the use of ever more advanced methods to meet the mission objectives. In a setting where we encounter short decision times or long distances between the vehicle and the operator, autonomy is essential. In defense applications a measurement must be quickly translated into a control command while spacecraft performing maneuvers near remote celestial bodies are beyond the communication threshold for manual command verification and require a capability for situational awareness to successfully meet their mission objectives.

Traditional navigation relies on dead-reckoning of inertial measurement unit (IMU) data for the state estimate during the phase of the entry where the atmospheric forces are greatest [15]. During this time the greatest adjustments to the entry trajectory can be performed as long as the spacecraft state is known to the accuracy necessary to make these adjustments. This phase of the hypersonic entry is therefore essential to achieve precision landing [6].

In regard to the problem of dead-reckoning compared to the use of the IMU data in a model-based filter, the filter approach provides a variety of benefits. First the model-based filter processing IMU data as external measurements is expected to be superior in performance as shown in this thesis. Only a theoretically perfect IMU with no bias and free of misalignment with very low measurement noise could compare to the model-based filter. But regardless of the performance, there are additional benefits from the use of a model-based filter for the state estimation purpose. Also a filter is capable of continuing to provide a high quality state estimate in the case of data drop out. Such event will cause the dead-reckoning process to diverge. Furthermore dead-reckoning does not provide an uncertainty measure at the time of parachute deploy which the filter based state estimation process provides. In addition to these benefits a model-based filter can be used to for the purpose of environmental parameter estimation during entry, descent and landing. Beyond the hypersonic entry phase a model-based filter can utilize additional measurements that then become available to improve the navigated state.

1.2 Guidance, Navigation & Control

In order to enable a vehicle to operate autonomously without a human operator it is necessary to implement an onboard GN&C system for the desired task. An autonomous system requires the ability to determine the vehicle state and perform necessary adjustments independently of human operator commands. For the task of computing a vehicle position and velocity, a variety of sensors are available, ranging from LORAN and GPS to modern radar, IMU, laser and optical guidance [2]. At Mars, GPS and LORAN are not available. It is not expected that GPS-like infrastructure will be available in the near future.

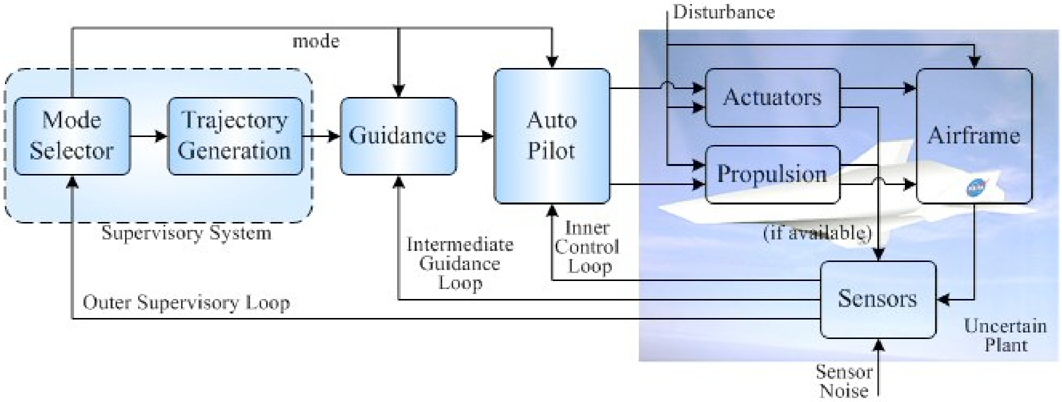

Figure 1.1: Structure of a Generic GN&C system.

As an integrated system, a GN&C system is used to determine the vehicle state, then determine and execute required trajectory corrections by commanding and controlling existing actuators. GN&C systems have a range of autonomy depending on the nature of the mission. The essential information of the vehicle state is provided by the navigation algorithm which combined with various control methods constitute the GN&C system as illustrated in Fig (1.1). Multiple tasks have to be performed. Inputs are provided by sensors, memory data, radio and satellite signals. The collected data is processed by a computer with one or multiple CPUs to calculate required control actions which are executed through the outputs that interact directly with the control hardware of the vehicle, such as propulsion and control surfaces.

Spacecraft GN&C systems have evolved significantly over the brief period of operational space flight. Today, high precision formation flying is performed which require sophisticated mathematical and computational efforts that are constantly being refined. As various GN&C architectures and methods exist, the required spacecraft control system has become one of the most complex systems of a mission and spans every phase of flight, from liftoff to landing.

The control aspect of the problem focuses on the mathematical modeling of a system, and using various elements of control theory to create the desired outcome. As linear control is well understood, real world situations rarely are captured comprehensively in a linear manner. To account for that a wide methodology of control ideas have been developed of which adaptivety and robustness play an essential role interplanetary spacecraft missions.

1.3 The Navigation Problem

The first step in a guidance and control system is to determine the state of the system in order to arrive at decisions about actions to be taken thereafter [26]. The process of obtaining this information is called 'navigation' [5]. A variety of means are available to find knowledge of the vehicle state and quality of this information. As for spacecraft navigation, this state usually is comprised of position, velocity and attitude. Dependent upon the specific phase of the mission design more or less complex models or state variables are used. In order to direct a spacecraft to a certain location, at the very least, the position and quality of the knowledge of the position must be considered.

Traditionally, the need for a navigated position has been realized in naval history for the determination of a ship's position on the open sea. Celestial bodies were used as the reference and dead reckoning provided a method to estimate a position between celestial updates. In modern times, triangulation and trilateration methods have been used, with the most recent development of Global Position Systems (GPS) that use spacecraft signals as their reference [27].

Ultimately the determination of the state requires application of physical methods in order to collect a measurement and mathematical tools to extract the information desired. Measurements utilized for this task are visual observations of the environment as the surrounding terrain or on a larger scale celestial bodies, or use of radio or satellite signals which are then processed and provide a knowledge of the state.

For the specific task of pinpoint landing at Mars, the trajectory segment passing through the upper atmosphere is essential as it is here that aerodynamic forces can be modulated and entry dispersions controlled. At the same time there are only few or no external measurements available for navigation at this point. The challenge is to maintain good knowledge of the spacecraft state in the presence of a highly dynamic environment in the presence of noise and unknown forces. It is here that the dead-reckoning navigation diverts from the actual spacecraft state and cannot meet the demands of pinpoint precision landing.

To address this situation, a model-based filter navigation architecture is proposed for pinpoint landing to achieve the required precision. This navigation method addresses the shortcoming of the dead-reckoning process by adding a dynamic model to the navigation process. By processing the heretofore dead-reckoned IMU measurements as external measurements in a model-based filter the navigation accuracy becomes more precise and robust.

1.4 A Brief History of Navigation

The forms and methods used for spacecraft navigation are a direct extension of traditional naval navigation and, more recently, aviation navigation methods, which were extended to meet space mission requirements that lead to a modern concept of navigation [5]. The skill of navigation became an issue for humans as they first set out to cross the oceans. Initially, navigation relied on one's senses, knowledge of nature and experience. Stars have always been a major guide for the task along with observation of the weather and known occurrences in nature. The weather, wildlife and color of the sea, sky and clouds were used to navigate until navigation became a major concern for commerce and military naval power when structured methods for navigation were developed.

Celestial navigation has always retained an importance, but has become a refined means of naval navigation. Further methods developed for the purpose are dead-reckoning, and, more recently, various methods of electronic navigation.

The roots of navigation can be found in nautical determination of position. With the development of vessels that were able to sail the oceans, the development of navigation tools became an important concern. Traditionally the latitude of a ship's position would be obtained from observation of stars or the sun. Prior to the use of calculus, tables of the sun were used in combination with a measurement of the angle of the sun over the horizon.

The determination of longitude was more complex [13]. It was calculated as a time difference between the same celestial event at different locations. In order to accomplish this task, an accurate clock was needed to be carried by the navigator. The difference between local noon and the reference clock gives the longitude. This method was refined for the navigator to use the Sun, Moon, Planet or any major stars for a position determination. The availability of an accurate time has remained an important part of navigation. Quartz watches, radio time broadcasts and atomic clocks are now in use for this purpose. Today navigation utilizes GPS as the primary reference.

With the development of aircraft, new navigation challenges appeared. The aircraft position must be determined while traveling at high speeds leaving less time to perform classical computations associated with nautical navigation. Aircraft cannot remain stationary in their position for the process of collecting measurements. A good knowledge of the position and velocity is essential as the aircraft can only remain in the air for a limited amount of time before it will have to stop for refueling. Besides the visible surroundings, instruments and radio navigation are used for navigation. For general aviation, the airspace is heavily controlled today and flight planning along with in-flight navigation and coordination between pilots and controllers manage the guidance and navigation purpose.

Outside the civilian operator controlled aviation sector, a variety of vehicles have very sophisticated systems that allow for autonomous operation. These are unmanned aerial vehicles and guided missiles. As they are utilized in many different situations, no comprehensive model can be provided to them and they have to perform well in different environments to successfully accomplish their tasks. This has led to highly complex GN&C systems in use today combining the most accurate navigational tools of our times.

The science of navigation was adjusted for space travel for the Apollo voyage to the moon. In order to successfully perform this task many concepts used in aeronautical and ship navigation had to be expanded for traveling in space. The concept to determine position by measuring angles between known objects as maintained and the various instruments developed for the navigation purpose though history had a modern counterpart on board the Apollo spacecraft. While the forces affecting the spacecraft on the way to the moon are well understood, the problem to be addressed was the situation of having a vehicle travel at high velocity. Only minor corrections to the course can be carried out as the energies required to alter a vehicle traveling at such high velocity are larger and only limited propellant is available for the complete mission. A good knowledge of the spacecraft state is essential to the success of the mission particularly for the return voyage from the moon to earth. The most crucial mission segment requiring accurate knowledge of the spacecraft state is the re-entry of the vehicle into the atmosphere of earth. At that instance a path too steep will result in deceleration forces that may be too high to withstand and a path too shallow can lead to the spacecraft skipping off the atmosphere [7].

The field of spacecraft navigation has many elements governed by the mission goal and mission segments, it is an extension of aircraft guidance with increased demands on versatility and autonomy. As spacecraft in earth orbit can be observed during most of their mission time, interplanetary vehicles need to have the ability to properly handle flight regimes they encounter without operator assistance.

For interplanetary transfer navigation the Deep Space Network (DSN) is used to regularly track the vehicle and update the trajectory when required. Upon arrival at the target, however, this tracking method cannot be used due to long signal travel times. For planetary entry, the time window from entry interface (EI) to touch down of several minutes requires the GN&C system to be able to independently determine the spacecraft state. For this task, the tool of the Kalman filter is being proposed. Based on IMU and other available data, the Kalman filter provides a best estimate of the state and a corresponding measure of the accuracy of the estimate. Based on this estimate the guidance and control can be adjusted to arrive at the desired landing site.

1.5 Model-Based Navigation

The navigation problem is addressed here utilizing a model-based navigation strategy to improve upon dead-reckoning navigation. The basic idea of model-based navigation is to predict the spacecraft state first and then use measurements to correct the prediction. The Kalman filter is the most successful model-based navigation method for spacecraft navigation. The extended Kalman Filter (EKF) has long been used for spacecraft navigation beginning with the Apollo missions [14].

To improve upon traditional navigation methods and achieve the precision required for pinpoint landing on Mars a new approach is required. As the EKF has long been used for the navigation task, a new variation of the Kalman filter, the sigma point Kalman filter (SPKF) is being proposed. During the hypersonic EDL where the only available information is obtained from the IMU, the SPKF will process IMU data as external measurements in combination with the model-based filtering algorithm. This will replace the dead reckoning of the IMU data during this phase. Also, a filter bank will be used to identify atmospheric parameters and pick the best set of model parameters throughout the EDL mission segment. The filter bank promises the adaptability to the navigation process.

This approach represents a novel architecture compared to the customary dead-reckoning of the IMU data for the navigation task. This process is expected to require more computational effort in the process of obtaining a state estimate, however the model based filter will resolve issues of pure dead-reckoning and will also provide a more accurate state estimate.

The adaptability of the filter to the experienced environment is essential for an entry into an uncertain atmosphere as encountered for a mission to the surface of Mars. As the atmosphere of Mars is highly dynamic and uncertain due to seasons and winds, an adaptive filter bank is essential for a pinpoint entry. While tuning a filter for a particular environment, is usual and customary, the uncertainty in the atmospheric conditions prevents a particularly tuned filter to operate optimally for an entry at Mars. The navigation filter will continuously have to adapt [24].

1.6 Relevance to Past, Present and Future Exploration Programs

Previous methods used for spacecraft entry were not created specifically for guided entry. In fact, a guided entry at a distant planet has never before been attempted. Traditionally a spacecraft entering the Mars atmosphere was rolled to reduce landing dispersions [23]. The Mercury and Gemini capsules had an offset between the center of mass and the center of pressure to allow for some degree of maneuverability. During the entry, the IMU and a horizon sensor provided navigation data. During Apollo [19], [20] acceleration-like measurements were used as inputs along with gravity models to propagate the position and velocity vector.

No Mars mission has attempted precision landing. The Viking [8] spacecraft used an inertial reference unit (IRU) with accelerometers and gyros, temperature and stagnation pressure sensors. As there was no entry guidance, the data from the instruments was mostly utilized for post-flight trajectory reconstruction.

The Pathfinder lander had no IMU and performed a ballistic entry. Sensors available were accelerometers. These determined the time to parachute deploy. Landers [3], [4] at other planets have similarly not had any active guidance between entry interface and parachute deploy [22], [25].

A variety of anticipated missions to Mars, such as the sample return missions will have to utilize precision navigation for their hypersonic entry in order to achieve the required landing accuracy. The architecture proposed in this work will be able to accommodate a large range of possible sensors during EDL to achieve the precision navigation capability needed. Drawing from established navigation tools used for Apollo, an SPKF-based navigation architecture seems best suited for the task [16]. The use of IMU data as external measurements is a step past existing technology and is expected to provide the navigation system with new capabilities, particularly during the hypersonic phase of EDL. The implications of this architecture for future missions are potentially significant as sufficient computational resources become available.

1.7 Contributions to the Field

The research results presented in this dissertation have a range of contributions to the field of spacecraft navigation. The SPKF for spacecraft precision navigation [16] utilizes the concept of nonlinear regression [17] as a new central computational tool in the filter bank [9]. This work is geared directly toward a guided entry at Mars which has never before been attempted [8], [19] and is an essential goal for NASA. It is the first contribution of this work to establish the SPKF for the purpose of spacecraft precision navigation during entry.

The idea of banks of Kalman filters has been developed and tested [28], [9]. A logical next step would be to use the SPKF as the computational unit in a filter bank, replacing the extended Kalman filter. Due to the characteristics of the filter, the gating network can be simplified while the filter continues to be capable of identifying the environment it is encountering.

A further contribution of this work is the implementation of the navigation filter architecture on the Electra programmable radio platform. In order to enable the concept to be utilized for the intended NASA missions, it is necessary that the SPKF be ported to the flight hardware. As this task has numerous constraints, the successful implementation of the SPKF on the Electra represents another contribution to the state-of-the-art.

1.8 Presentation of Work

Chapter 2 provides an in-depth discussion of the Kalman filter and prepares the reader for the concept of the SPKF which is the underlying tool used for the proposed navigation architecture. After a brief history of the Kalman filter and a review of the filter equations, the modification of the concept for the EKF is given and the newly developed concept of the SPKF is presented. The different estimation methods are then applied to a simple example and the quality of the different concepts is compared. Then, the architecture of a filter bank is introduced and the filter banks gating network is described. And validated using an example that has been subject to the spacecraft precision navigation analysis presented in detail in Chapter 5.

Chapter 3 introduces the entry challenges which this work has considered. For most of the work, the Mars Exploration Rover (MER) mission was used to establish the validity of the SPKF for the task. As actual mission data is available from the MER EDL, this mission is intensely studied and analyzed for the task. As a future navigation challenge, the Mars Science Laboratory mission is discussed then and presented as a possible target for the proposed architecture to be validated as an in-flight navigation tool.

The underlying mathematical models are described in Chapter 4. The EDL problem presents several modeling challenges. The spacecraft dynamics are the first set of models described. The environmental models are then discussed and defined. Uncertainties in knowledge of gravity and atmosphere parameters are considered. Furthermore, the measurement model is presented and simulated measurement histories obtained for MER and MSL. This chapter concludes with a discussion of the simulated trajectories used for MER and MSL that become the reference for the filter performance analysis. The description of the process of dead reckoning of the measurements is presented, as this method is traditionally the only way to obtain a state estimate for the hypersonic entry phase.

The results of the simulations are discussed in Chapter 5 for MER and MSL. The performance of the SPKF is analyzed for various levels of noise encountered and the applicability of the SPKF for the purpose of hypersonic EDL is established. As an advanced investigation, a SPKF filter bank is evaluated for environmental parameter estimation of the atmospheric density encountered during EDL. With a filter bank containing a number of different atmospheric models, the filter bank is shown to be capable of determining the environment actually encountered.

An important part of this work is the implementation of the filter architecture on the Electra programmable radio hardware. The hardware implementation is discussed in Chapter 6. A description of the Electra platform is followed by the discussion of the Electra emulator as a simulation environment for flight software development. Several issues that had to be addressed during the implementation of the architecture for the hardware are discussed and the solutions presented. The performance of the architecture on the hardware concludes this chapter.

Finally, the results of this work lead to conclusions and intended future work which is presented in Chapter 7. Additional data that was used in the analysis of various trajectories that were subject of this work are presented in Appendix A.

Chapter 2

Filtering Strategy

From the beginning of human space travel, it became paramount to have a method to accurately determine the spacecraft state, including position, velocity and attitude. The Kalman filter was developed for the Apollo program and has since become the backbone of spacecraft navigation systems [19].

Over time various updates have been made to adjust the navigation concept to the needs of modern spacecraft and computing systems. In the process of improvement of navigation, variations of the Kalman filter were developed. The Kalman filter is the tool upon which the proposed navigation concept is based.

The Kalman filter has undergone significant changes over time to make the concept adaptable to a variety of applications. The need to place the filter in real-time operating environments lead to many engineering solutions that attempt to make the filter concept more versatile and implementable with less computational overhead. A computationally efficient and mathematically uncomplicated version of the Kalman filter was developed and is represented by the sigma point Kalman filter (SPKF) [28]. This approach is chosen as the main computational unit in this work.

2.1 Brief History of the Kalman Filter

With the advent of the jet age and supersonic flight, aircraft required advanced flight control mechanisms. Thus, in the late 1950s and early 1960s, the Air Force Office of Scientific Research (AFOSR) sponsored various research efforts in the area of control theory related to high-speed aircraft, aerospace vehicle systems, and advanced space systems. Designers had a growing need to maintain aerodynamic integrity by deriving the best information out of numerous and continuous data streams that inherently contained imprecise data [1].

AFOSR sponsored many efforts in this general area, including pioneering support of research in nonlinear systems and especially the support given to the Research Institute of Applied Science (RIAS). One such AFOSR-sponsored RIAS program involved the development and application of statistical filtering theory under the direction of Dr. Rudolph E. Kalman and Dr. Richard Bucy. AFOSR initiated this support to investigate the use of modern mathematical statistical methods in estimation. AFOSR program managers perceived this as an opportunity for the creation of new mathematical techniques might significantly alter control applications. With AFOSR support, Kalman and Bucy wrote several seminar papers that ultimately led to the development of the Kalman filter. This invention revolutionized the field of estimation and had an enormous impact on the design and development of precise navigation systems. The Kalman and Bucy technique of combining and filtering information from multiple sensor sources achieved accuracies that clearly constituted a major breakthrough in navigation technology.

Dr. Kalman's innovative algorithm [14] eventually made the transition from a relatively abstract theory to practical application, especially for the nation's space program. The National Aeronautics and Space Administration (NASA) embraced the Kalman filter to solve the problems associated with determining satellite orbits. It quickly became a basic building block of the space program and was first used in the Ranger, Mariner, and Apollo missions of the 1960s. The Kalman filter quickly found its way into other operational systems such as phased-array radars to track missiles, inertial guidance systems in aircraft (from the C-5 to the F-22), submarines, missile autopilots, the Global Positioning System and the Space Shuttle.

Almost all modern control systems, both military and commercial, use the Kalman filter. Although the extended Kalman filter (EKF), which incorporates nonlinear state estimation, is not theoretically-vigorous, the EKF proves to be a tractable, highly successful design and is integrated into many aerospace systems.

The SPKF has been discussed in the literature [17], [28] as an alternative to the EKF for use in nonlinear filtering applications. While retaining the structure of the recursive filtering logic associated with the EKF, the SPKF employs a statistical transformation in a recursive filter structure that arguably provides more accurate estimates, while offering architectural benefits and requiring similar computational resources.

A SPKF has not previously been used for entry, descent and landing (EDL) navigation. Numerous types of EKFs have been applied to the EDL problem in various architectures and studied extensively. The EKF navigation algorithms and have worked quite well overall in space applications. Ultimately, however, our goal is to develop an implementation of an adaptive navigation algorithm based on the multiple-model architecture that can be operated in real-time with limited computational resources. To this end, we are striving to "fit" our navigation algorithms into the Mars Program Office Electra programmable radio. The SPKF offers a simplified filter structure due to the fact that the algorithm is derivative free. No linearization of the dynamic equations is required enabling the use of high quality models not reasonably implementable in the EKF. Similarly, with the SPKF a measurement model is used in the estimation process without the requirement to obtain potentially complicated measurement partials associated with the measurement mapping matrix. In this implementation of the SPKF, measurements used are IMU data. Doppler ranging and velocimeter measurements become available after heat shield jettison, while IMU data can be processed from the beginning of the entry phase at entry interface (EI). This work focuses on the pre-heat shield jettison phase.

2.2 Overview of the Kalman Filter

The Kalman filter processes external measurements to compute an estimate of a state. Using a recursive filter structure, the process has two main

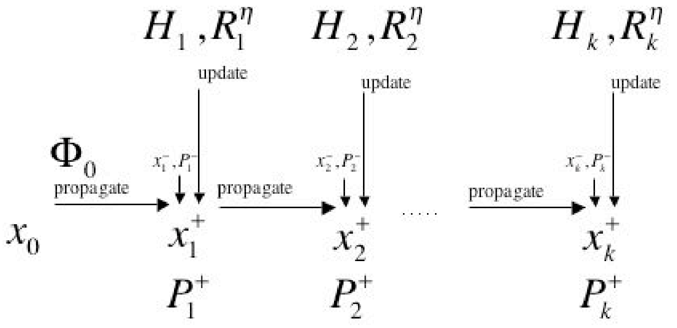

Figure 2.1: Kalman Filter Timing Diagram.

steps: propagate and update, as illustrated in Figure (2.1). More information about the Kalman filter can be found in [14]. The Kalman filter is model based, that is, the filter architecture is based on an assumed model of the dynamics and the measurements.

The model is assumed to have the form

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_{k+1}\mathbf{x}_k + \mathbf{r}_{\nu,k} \tag{2.1}$$

where $\mathbf{x}_k \in \Re^n$ is the state vector, Φ_k is the state transition matrix and $\mathbf{r}_{\nu,k}$ is the process noise. The process noise is assumed to be a zero-mean, white noise sequence with $E[\mathbf{r}_{\nu,k}\mathbf{r}_{\nu,j}^T] = \mathbf{R}_j^\nu \delta_{jk}$.

The measurement is modeled as

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{r}_{\eta,k} \tag{2.2}$$

where $\mathbf{z}_k \epsilon \Re^m$ are the measurements, $\mathbf{H}_k$ is the measurement mapping matrix, and $\mathbf{r}_\eta$ is the measurement noise. The measurement noise is assumed to be a zero-mean, white sequence with $E[\mathbf{r}_{\eta,j}\mathbf{r}_{\eta,k}^T] = \mathbf{R}_j^\eta \delta_{jk}$.

At the update stage when a measurement is available, the Kalman filter state update is

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k[\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-] \tag{2.3}$$

where $\hat{\mathbf{x}}_k^-$ is the state estimate before the update and $\hat{\mathbf{x}}_k^+$ is the state update after the update. The Kalman gain is computed via

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T[\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k^\nu]^{-1}. \tag{2.4}$$

The state estimate error covariance is updated via

$$\mathbf{P}_k^+ = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{P}_k^-. \tag{2.5}$$

Between measurements, the state estimate and state estimation error covariance are propagated via

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{\Phi}_k\hat{\mathbf{x}}_k^+, \tag{2.6}$$

and

$$\hat{\mathbf{P}}_{k+1}^- = \mathbf{\Phi}_k\mathbf{P}_k^-\mathbf{\Phi}_k^T + \mathbf{R}_k^\eta, \tag{2.7}$$

respectively.

2.3 The Extended Kalman Filter

The Kalman filter is suited for linear models and provides an optimal estimate in that case only. Most real problems contain nonlinearities. It is therefore necessary to find a way to integrate nonlinear models into the Kalman filter structure. This problem can be addressed by the extended Kalman filter (EKF).

Based on the Kalman filter, the EKF is an *ad hoc* filter. While the EKF retains the same architecture (propagate and update) as the Kalman filter, there is very little supporting theory to guarantee stability and optimality.

The dynamics are modeled via

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, t) + \mathbf{r}_\nu(t) \qquad t_{i-1} \leq t \leq t_i \tag{2.8}$$

with $\mathbf{r}_\nu(t)$ is a zero-mean, white noise with $E[\mathbf{r}_\nu(t)\mathbf{r}_\nu^T(\tau)] = \mathbf{Q}(t)\delta(t-\tau)$. The measurements at time t_i are assumed to have the form

$$\mathbf{y}_i = \mathbf{h}(\mathbf{x}(t_i)) + \mathbf{r}_{\eta,i} \qquad t = t_i \tag{2.9}$$

where $\mathbf{r}_{\eta,i}$ as a zero-mean white sequence with $E[\mathbf{r}_{\eta,j}\mathbf{r}_{\eta,k}^T] = \mathbf{R}_j^\eta \delta_{jk}$. Taking the expectation of Eq.(2.8) and interchanging the expectation and differentiation operator yields

$$\frac{d}{dt}E[\mathbf{x}(t)] = E[f(\mathbf{x}(\mathbf{t}), t)]. \tag{2.10}$$

This can be written as

$$\dot{\hat{\mathbf{x}}}(t) = \hat{f}(\mathbf{x}, t). \tag{2.11}$$

where the “hat” notation denotes expectation . The initial condition is selected to be $\hat{\mathbf{x}}(t_{i-1}) = E[\mathbf{x}(t_{i-1})]$. With the state estimation error covariance defined as

$$\mathbf{P}(t) \equiv E[(\hat{\mathbf{x}}(t) - \mathbf{x}(t))(\hat{\mathbf{x}}(t) - \mathbf{x}(t))^T] \tag{2.12}$$

it follows that

$$\dot{\mathbf{P}}(t) = \mathbf{x}(t)\widehat{\mathbf{f}^T(\mathbf{x}(t)}, t) - \hat{\mathbf{x}}(t)\hat{\mathbf{f}}^T(\mathbf{x}(t), t) + \mathbf{f}(\mathbf{x}\widehat{(t), t)\mathbf{x}}^T(t) - \hat{\mathbf{f}}(\mathbf{x}(t), t)\hat{\mathbf{x}}^T(t) + \mathbf{Q}(t).$$

Expanding $\mathbf{f}(\mathbf{x}, t)$ in a Taylor series about the current estimated state yields

$$f(\mathbf{x}, t) = f(\hat{\mathbf{x}}, t) + \frac{\partial f}{\partial \mathbf{x}}|_{\mathbf{x}=\hat{\mathbf{x}}}(\mathbf{x} - \hat{\mathbf{x}}) + \ldots \tag{2.13}$$

Taking the expectation and dropping higher-order terms in Eq. (2.13), it follows that

$$\hat{f}(\mathbf{x}(t), t) \cong f(\hat{\mathbf{x}}(t), t). \tag{2.14}$$

where we use the fact that $E(\mathbf{x} - \hat{\mathbf{x}}) = 0$ (i.e., an unbiased estimate).

Then, we have

$$\dot{\hat{\mathbf{x}}}(t) \cong f(\hat{\mathbf{x}}(t), t) \qquad t_{i-1} \leq t \leq t_i \tag{2.15}$$

and

$$\dot{\mathbf{P}}(t) \cong \mathbf{A}(\hat{\mathbf{x}}(t), t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{A}^T(\hat{\mathbf{x}}(t), t) + \mathbf{Q}(t) \tag{2.16}$$

where

$$\mathbf{A}(\hat{\mathbf{x}}(t), t) \equiv \frac{\partial f(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)}|_{\mathbf{x}=\hat{\mathbf{x}}(t)}. \tag{2.17}$$

The EKF assumes a linear update of the form

$$\hat{\mathbf{x}}_i^+ = \hat{\mathbf{x}}_i^- + \mathbf{K}_i(\mathbf{y}_i - \hat{\mathbf{h}}(\mathbf{x}(t_i))). \tag{2.18}$$

Making the same assumptions (that is, expand $\mathbf{h}(\mathbf{x}(t_i))$ about $\mathbf{h}(\hat{\mathbf{x}}(t_i))$ yields to first order

$$\hat{\mathbf{h}}(\mathbf{x}(t_i)) \cong \mathbf{h}(\hat{\mathbf{x}}_i^-) \tag{2.19}$$

.

The update equation the becomes

$$\hat{\mathbf{x}}_i^+ \cong \hat{\mathbf{x}}_i^- + \mathbf{K}_i(\mathbf{y}_i - \mathbf{h}(\hat{\mathbf{x}}(t_i)). \tag{2.20}$$

In summary, the EKF propagation is accomplished via

$$\dot{\hat{\mathbf{x}}}(t) = f(\hat{\mathbf{x}}(t), t) \qquad t_{i-1} \leq t \leq t_i \tag{2.21}$$

$$\dot{\mathbf{P}}(t) = \mathbf{A}(\hat{\mathbf{x}}(t), t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{A}^T(\hat{\mathbf{x}}(t), t) + \mathbf{Q}(t) \qquad t_{i-1} \leq t \leq t_i$$

and $\hat{\mathbf{x}}(t_{i-1}) = \hat{\mathbf{x}}_{i-1}^+$ and $\mathbf{P}(t_{i-1}) = \mathbf{P}_{i-1}^+$. The update is accomplished with

$$\hat{\mathbf{x}}_i^+ = \hat{\mathbf{x}}_i^- + \mathbf{K}_i(\mathbf{y}_i - \mathbf{h}(\hat{\mathbf{x}}(t_i)) \tag{2.22}$$

$$\mathbf{P}_i^+ = (\mathbf{I} - \mathbf{K}_i\mathbf{H}_i(\hat{\mathbf{x}}_i^-))\mathbf{P}^- \tag{2.23}$$

$$\mathbf{K}_i = \mathbf{P}_i^-\mathbf{H}_i^T(\hat{\mathbf{x}}_i^-)[\mathbf{H}_i(\hat{\mathbf{x}}_i^-)\mathbf{P}_i^-\mathbf{H}_i^T(\hat{\mathbf{x}}_i^-) + \mathbf{R}_i]^{-1} \tag{2.24}$$

with

$$\mathbf{A}(\hat{\mathbf{x}}(t), t) \equiv \frac{\partial f(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)}|_{\mathbf{x}=\hat{\mathbf{x}}(t)} \tag{2.25}$$

and

$$\mathbf{H}_i(\hat{\mathbf{x}}_i^-) \equiv \frac{\partial h(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)}|_{\mathbf{x}=\hat{\mathbf{x}}_i^-}. \tag{2.26}$$

2.4 The Sigma Point Kalman Filter

Various technical issues surrounding the SPKF [15], [16] are the subject of this investigation. The first issue of concern is the application of the SPKF as an alternative to the EKF for real-time navigation during EDL of an actively guided spacecraft at a distant planet with a sensible atmosphere. A particular implementation of the SPKF is proposed and studied. The second area of investigation addresses a separate, but related, issue of the manner in which inertial measurement unit (IMU) data is incorporated in the navigation system. Spacecraft navigation systems traditionally employ EKFs to process external measurements to compute state updates at discrete times while being aided by an IMU providing measurements of acceleration and attitude between measurements. Here we consider the alternative strategy of incorporating models of the IMU and aerodynamics into the filter so that the IMU data are processed as if they were external measurements.

As in the case of the EKF, the SPKF propagates the state between measurements based on a nonlinear (and often complex) dynamic model and measurement model dynamics to obtain a new *apostereori* estimate. The background of the filter theory is the decomposition of the initial state and covariance into a number of sigma points governing the propagation and update. In this implementation, the filter state estimate and estimation error covariance are used to create a collection of representative vectors with the same statistical properties as the state estimate and estimation error covariance. These equivalent state vectors are referred to as sigma points with an assigned weight for each vector. For the state propagation step, the individual vectors are propagated through the complete nonlinear dynamic model of the system. Based on the assigned weights a new propagated state is obtained.

This purely statistical method of propagating the filter state vector and covariance matrix avoids the need to obtain a partials matrix for the filter.That is there is no need to explicitly compute $\mathbf{A}(\hat{\mathbf{x}}(t))$ and $\mathbf{H}_i(\hat{\mathbf{x}}_i^-)$ in Eqs.(2.25) and (2.26), respectively. It has been demonstrated in this work that the quality of this propagation method is superior to the EKF for nonlinear dynamics [28].

The measurement update follows a similar strategy using the sigma points. For each individual sigma point, an associated estimated measurement is obtained from the measurement model. Based on this collection of representative measurements, the new measurement covariance and cross-covariance are obtained that are in turn used to calculate the Kalman gain to obtain the *apostereori* state estimate and state estimation covariance matrix.

The SPKF is based on the Unscented Transformation (UT). The UT is applied to obtain the mean and covariance of stochastic variables given a set of measurements and a measurement model. In order to arrive at the mean and covariance of a variable, a set of weights is required. Utilizing the concepts and coding architectures outlined in [15], the SPKF [28] works as follows:

Given an initial mean and a covariance

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}] \tag{2.27}$$

$$\mathbf{P}_0 = E[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T] \tag{2.28}$$

the set of sigma points is obtained by Cholesky decomposition of the covariance matrix and combining with the mean according to

$$\chi_{k-1} = [\hat{\mathbf{x}}_{k-1} \quad \hat{\mathbf{x}}_{k-1} \pm (\sqrt{(L+\lambda)\mathbf{P}_{k-1}})_i], \tag{2.29}$$

with $\sqrt{\mathbf{P}_{k-1}}$ as the $k-1$st column of the Cholesky decomposition. The total number of sigma points is $2L+1$ where $\chi \, \epsilon \, \Re^{2L+1}$, and with $\bar{\mathbf{x}} = E[\mathbf{x}_{k-1}]$

$$\chi_0 = \bar{\mathbf{x}}, \tag{2.30}$$

$$\chi_i = \bar{\mathbf{x}} + \sqrt{((L+\lambda)\mathbf{P}_{k-1})_i} \tag{2.31}$$

for $i = 1 \ldots L$ and

$$\chi_i = \bar{\mathbf{x}} - \sqrt{((L+\lambda)\mathbf{P}_{k-1})_{i-L}} \tag{2.32}$$

for $i = L+1 \ldots 2L$.

For the propagation and update stages of the SPKF, each sigma point state and state estimation covariance matrix is assigned a weight according to

$$W_0^{(m)} = \frac{\lambda}{(L+\lambda)}, \tag{2.33}$$

$$W_0^{(c)} = \frac{\lambda}{(L+\lambda)} + (1 - \alpha^2 + \beta), \tag{2.34}$$

respectively, and

$$W_i^{(m)} = W_i^{(c)} = \frac{\lambda}{2(L+\lambda)}. \tag{2.35}$$

These weights contain tuning factors λ, α and β, satisfying

$$\lambda = \alpha^2(L - \kappa) - L, \tag{2.36}$$

where $\kappa = 0$, $\alpha = 10^{-3}$ (or a small value such as $1 > \alpha > 10^{-4}$), and β is usually set to 0 (generic probability distribution) or 2 (Gaussian distribution).

The SPKF propagation step requires the propagation of the sigma points. The sigma points are propagated through the nonlinear dynamics function $\mathbf{f}$ with process noise covariance $\mathbf{R}_\nu$ according to

$$\chi_{k|k-1} = \mathbf{f}(\chi_{k-1}, \mathbf{R}_\nu) \qquad t_{k-1} \leq t \leq t_k. \tag{2.37}$$

The propagated state and covariance are obtained by combining the set of propagated sigma points with their appropriate weights yielding

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \chi_{k|k-1} \tag{2.38}$$

and

$$\mathbf{P}_k^- = \sum_{i=0}^{2L} W_i^{(c)} E[(\chi_{k|k-1} - \hat{\mathbf{x}}_k^-)(\chi_{k|k-1} - \hat{\mathbf{x}}_k^-)^T]. \tag{2.39}$$

The SPKF update step is done based on a modeled measurement for each sigma point. For the measurement update, a measurement model $\mathbf{H}$ is applied to the set of propagated sigma points as

$$\upsilon_{k|k-1} = \mathbf{H}(\chi_{k|k-1}) \qquad\qquad t = t_k \tag{2.40}$$

Then, a reference measurement is computed according to the sigma point weights via

$$\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \upsilon_{i,k|k-1}. \tag{2.41}$$

This enables the calculation of the Kalman gain through the measurement covariance and the cross covariance, with $\mathbf{R}_\eta$ as the measurement noise covariance,

$$\mathbf{P}_{yy} = \sum_{i=0}^{2L} W_i^{(c)} E[(\upsilon_{i,k|k-1} - \hat{\mathbf{y}}_k^-)(\upsilon_{i,k|k-1} - \hat{\mathbf{y}}_k^-)^T] + \mathbf{R}_\eta \tag{2.42}$$

and

$$\mathbf{P}_{xy} = \sum_{i=0}^{2L} W_i^{(c)} E[(\chi_{i,k|k-1} - \hat{\mathbf{x}}_k^-)(\upsilon_{i,k|k-1} - \hat{\mathbf{y}}_k^-)^T]. \tag{2.43}$$

With the measurement, $\mathbf{y}$, the update equations for the filter are

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k^-) \tag{2.44}$$

and

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_{yy} \mathbf{K}_k^T \tag{2.45}$$

where the Kalman gain is calculated through

$$\mathbf{K}_k = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{y}_k - \hat{\mathbf{y}}_k^-)^T] E[(\mathbf{y}_k - \hat{\mathbf{y}}_k^-)(\mathbf{y}_k - \hat{\mathbf{y}}_k^-)^T]^{-1} \tag{2.46}$$

and

$$\mathbf{K}_k = \mathbf{P}_{xy} \mathbf{P}_{yy}^{-1}. \tag{2.47}$$

2.5 One-Dimensional Tracking Problem

A simple problem of a mass falling in a gravity field with no horizontal velocity component is used to analyze the performance of the different filters for the estimation purpose [14]. With known initial conditions, the estimated parameters are position, velocity and ballistic coefficient.

The state vector is defined as

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \\ \beta \end{bmatrix}. \tag{2.48}$$

The system dynamics are represented by the time derivative of the state represented as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = \begin{bmatrix} x_2 \\ d - g \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix}. \tag{2.49}$$

The drag deceleration is given by

$$d = \frac{\rho x_2^2}{2x_3}. \tag{2.50}$$

The atmospheric density is computed as

$$\rho = \rho_0 \exp^{-\frac{x_1}{K_p}} \tag{2.51}$$

where the atmospheric decay coefficient K_p, is $K_p = 22000ft$ and the reference density of the atmosphere $\rho_0 = 3.4 \times 10^{-3} \frac{lbs^2}{ft^4}$.

The gravity is assumed constant with $g = 32.2\frac{ft}{s^2}$.

The measurement is assumed to be the position perturbed by noise. The noise is assumed to be a sequence of zero-mean, white noise samples $r_\nu(t_i)$.

$$z(t_i) = x_1(t_i) + r_\nu(t_i). \tag{2.52}$$

The noise standard deviation for the measurement is

$$R = 100\frac{ft^2}{Hz} \tag{2.53}$$

The initial state estimate covariance is given by

$$\mathbf{P}_0 = \begin{bmatrix} P_{11_0} & 0 & 0 \\ 0 & P_{22_0} & 0 \\ 0 & 0 & P_{33_0} \end{bmatrix} \tag{2.54}$$

with initial values of $P_{11_0} = 500ft^2$, $P_{22_0} = 2 \times 10^4 \frac{ft^2}{s^2}$, and $P_{33_0} = 2.5 \times 10^5 \frac{lb^2}{ft^4}$.

The initial conditions are $x_1(0) = 10^5 ft$, $x_2(0) = -6000\frac{ft}{s}$, and $x_3(0) = 2000\frac{lb}{ft^2}$.

The equations of motion in Eq. (2.49) are

$$\dot{\mathbf{x}} = f(\mathbf{x}) = \begin{bmatrix} x_2 \\ d - g \\ 0 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{\rho x_2^2}{2x_3} - g \\ 0 \end{bmatrix}. \tag{2.55}$$

It follows

$$\mathbf{A}(\hat{\mathbf{x}}) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\hat{\mathbf{x}}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & \frac{\rho x_2}{x_3} & -\frac{\rho x_2^2}{2x_3^2} \\ 0 & 0 & 0 \end{bmatrix}|_{\mathbf{x}=\hat{\mathbf{x}}}. \tag{2.56}$$

With the measurement model in Eq. (2.52) it follows that

$$\mathbf{H}_i = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\hat{\mathbf{x}}} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}. \tag{2.57}$$

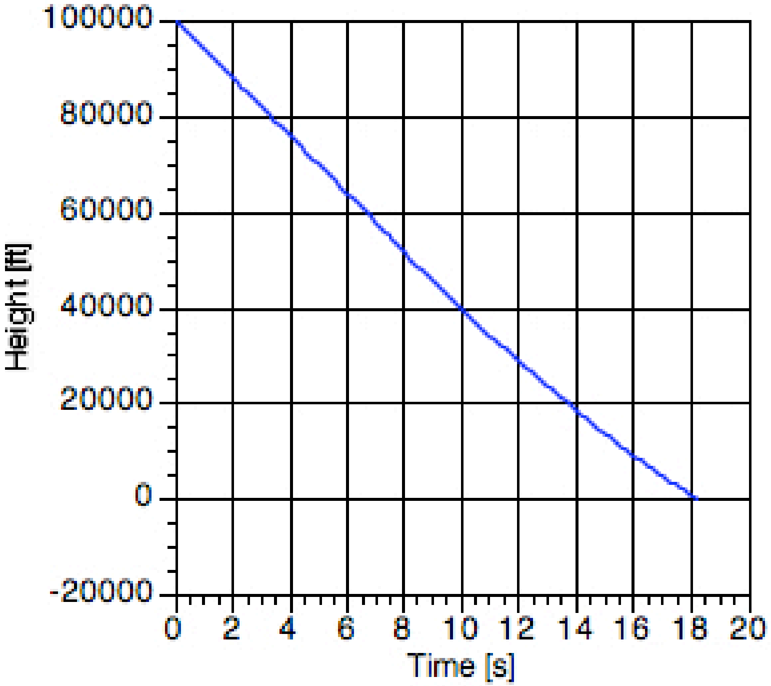

Figure 2.2: True Position of Falling Mass.

With this problem as a test case, the performance of the Kalman filter in a nonlinear dynamic system, the EKF and the SPKF are compared. The true history of the falling mass is presented in Fig. (2.2) and Fig. (2.3).

The results presented compare the estimates of position, velocity and the ballistic coefficient for the Kalman filter, the EKF, and the SPKF. The estimation error is represented by the blue line, the update at each time step is the yellow line. Also, the *apriori* covariance is shown as the green line while the *apostereori* covariance is the red line.

As expected the Kalman filter does not estimate the state well. This

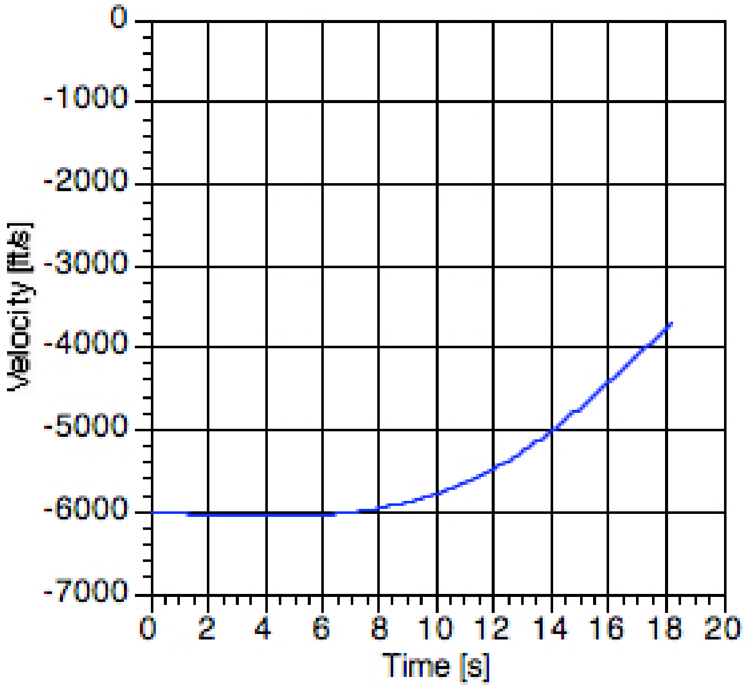

Figure 2.3: True Velocity of Falling Mass.

is due to the model nonlinearities. The Kalman filter estimates are shown in Fig. (2.4) through Fig. (2.6).

This situation is resolved through the use of the EKF. The EKF is able to estimate the state well and the uncertainty of the knowledge decreases over time with more measurements. The EKF results are shown in Fig. (2.7) through Fig. (2.9).

The SPKF estimates of the state, shown in Fig. (2.10) through Fig. (2.12), show good agreement with the true trajectory and the uncertainty decreases with time. It can be observed that the SPKF gives a better result when com-

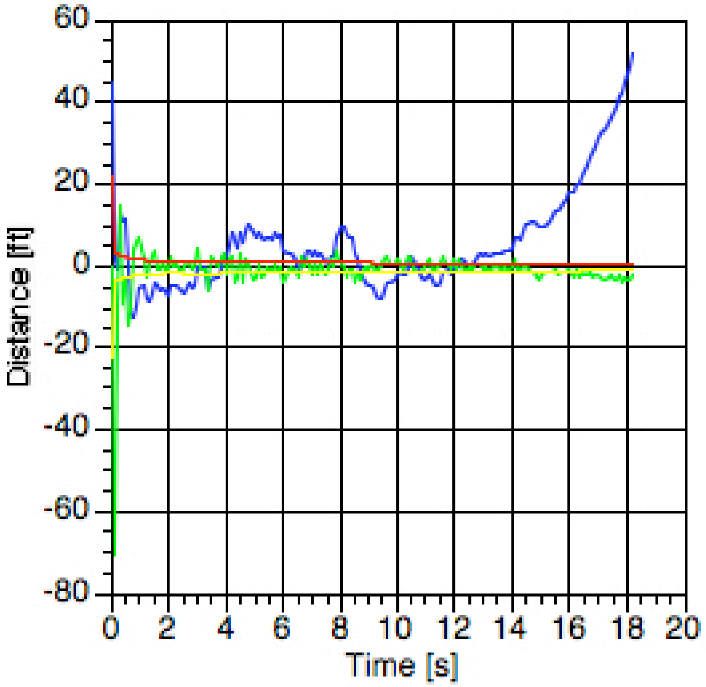

Figure 2.4: Kalman Filter Position Estimation Errors.

pared to the EKF. While the trend is similar, the SPKF obtains a higher quality estimate faster.

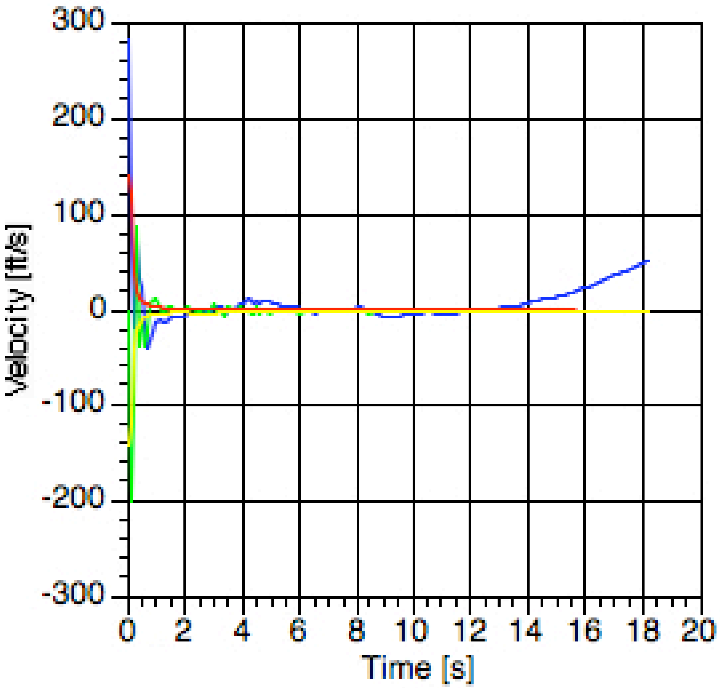

Figure 2.5: Kalman Filter Velocity Estimation Errors.

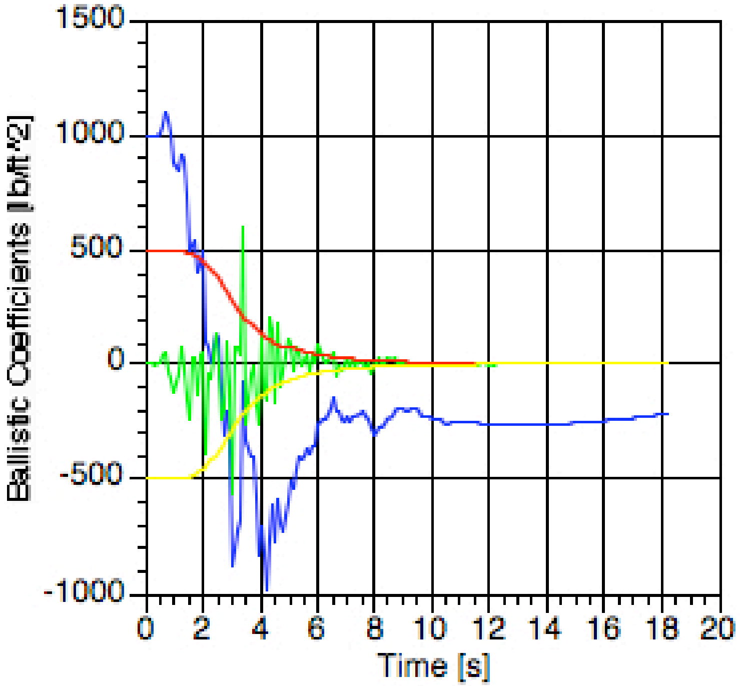

Figure 2.6: Kalman Filter Ballistic Coefficient Estimation Errors.

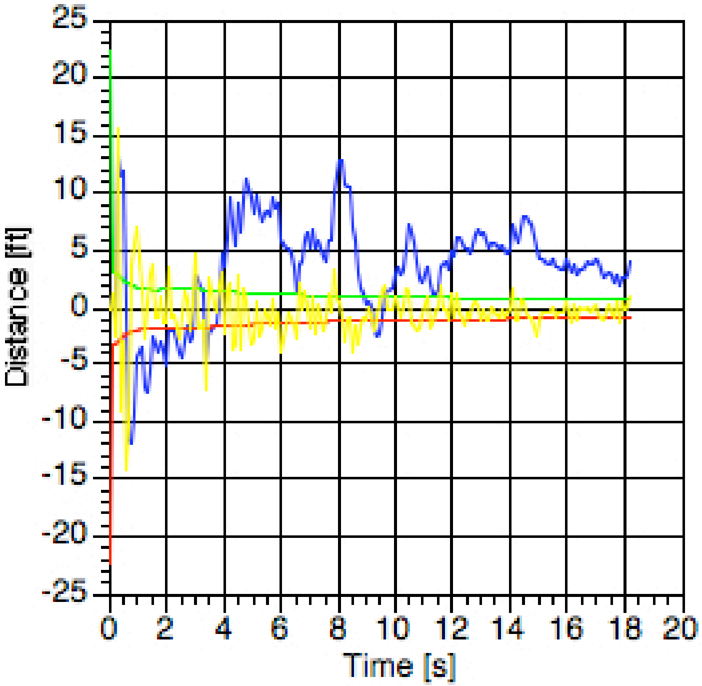

Figure 2.7: EKF Position Estimation Errors.

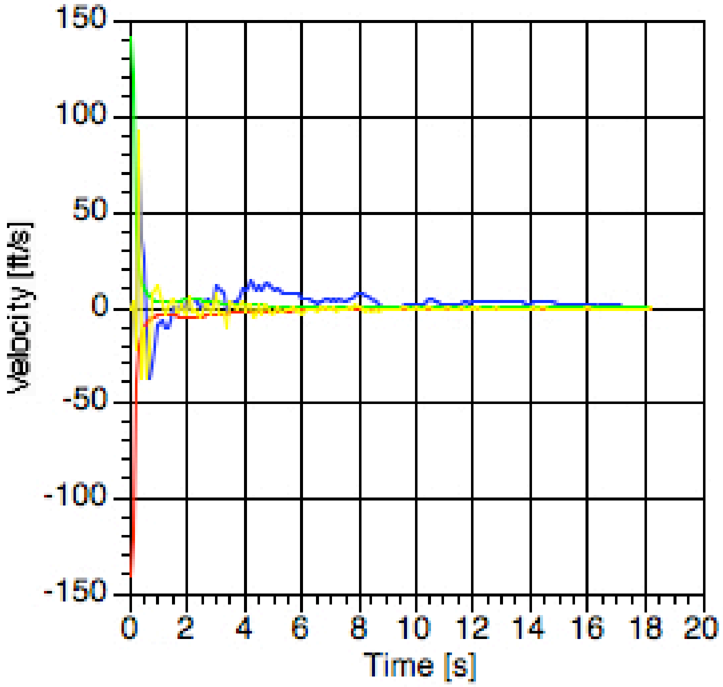

Figure 2.8: EKF Velocity Estimation Errors.

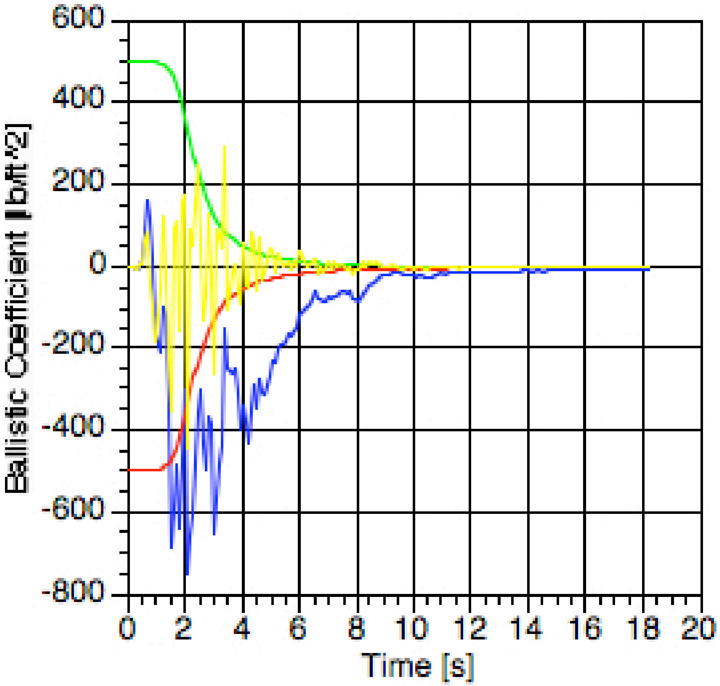

Figure 2.9: EKF Ballistic Coefficient Estimation Errors.

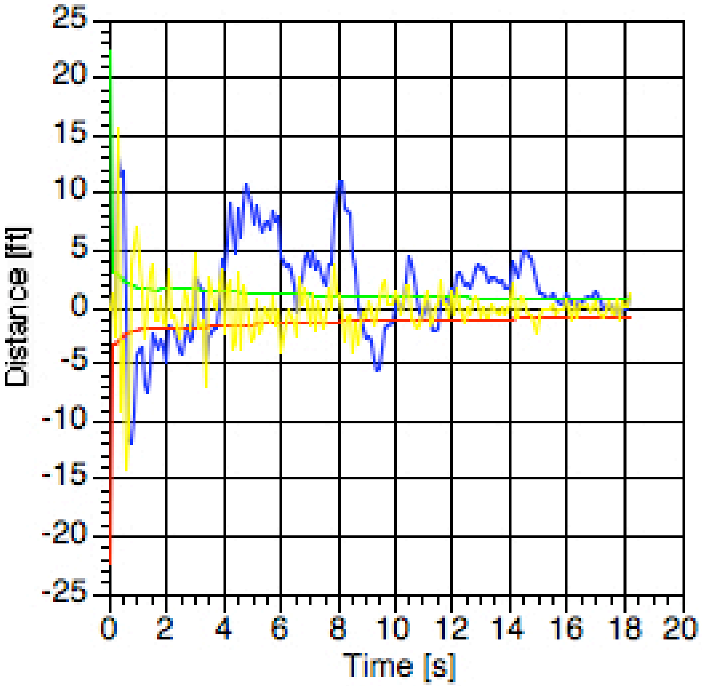

Figure 2.10: SPKF Position Estimation Errors.

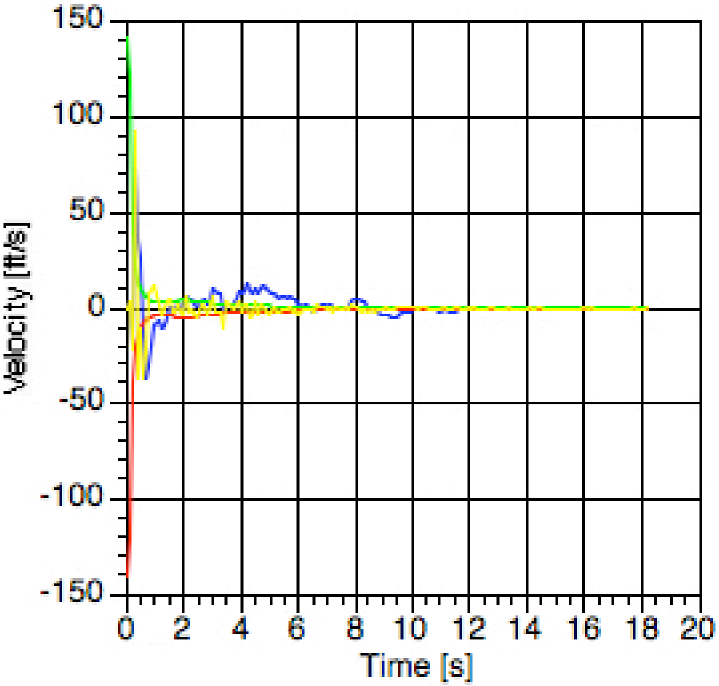

Figure 2.11: SPKF Velocity Estimation Errors.

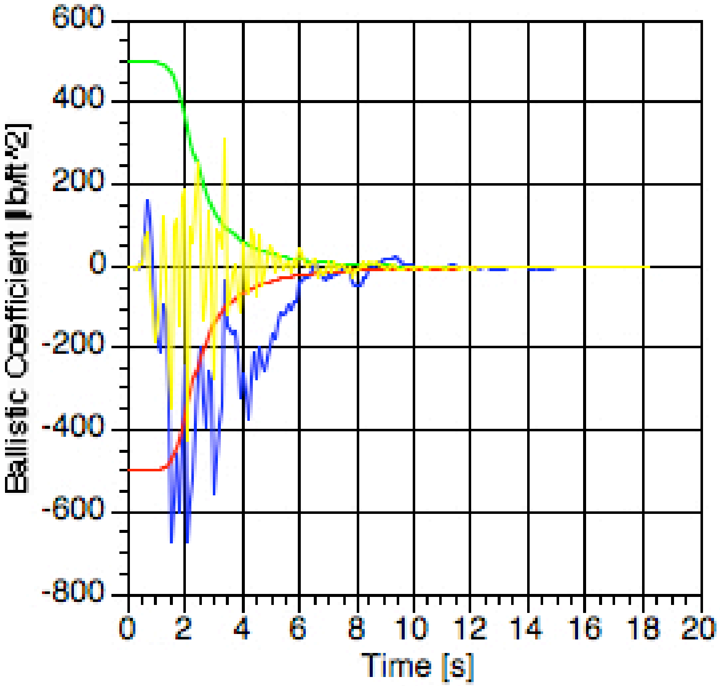

Figure 2.12: SPKF Ballistic Coefficient Estimation Errors.

2.6 Navigation Filter Architecture

Utilizing the benefits of the SPKF, the proposed navigation filter architecture will use the SPKF for the task of state estimation during EDL at Mars. At entry interface (EI), the navigation task will be performed by an operational SPKF that provides state and covariance during EDL. This filter will be tuned to the expected environment. It is proposed to use a bank of multiple filters to identify unexpected environmental conditions.

2.6.1 The Filter Bank

Upon implementation of a single filter, expansion to have multiple filters in a bank is straightforward. Multiple filters with a variety of parameter settings can be implemented in a bank and the best filter selected using a gating network.

Each of the filters will provide a state estimate and estimation error covariance. The challenge is how to identify which filter is performing best at each moment. This issue is addressed here and in Section 2.6.2.

A filter bank is comprised of multiple filters. We propose to have one *operational* filter that provides state estimate and estimation error covariance, and a set of filters that are used to identify the correct parameters to be used in the operational filter. A set of filters is implemented for the use of identifying the proper atmospheric model. The best working atmosphere is associated with the filter in the bank that is performing best. The atmospheric parameters of that selected filter can be used in the operational filter. It has been shown that the parameters are successfully chosen and the filter will settle on the most appropriate parameter settings during EDL.

The gravity model is a natural candidate to be included in the filter bank. To do this, a set of filters using the optimal settings in the other parameters must be implemented while various gravity models are modeled in this set of filters. From analysis of the measurements, the proper gravity model is then chosen in the same way the atmospheric model has been chosen in the initial set described. The proposed gravity filter bank would run simultaneously to the atmospheric filter bank.

2.6.2 The Gating Network

The gating network implemented in this particular filter bank is straight forward. For the purpose of choosing the proper atmospheric model, the IMU information is used as a measurement and each filter predicted measurement is compared against the IMU data. Basically the filter parameters selected (and used in the operational filter) are the values used in the filter that has the smallest sum of the squares of the difference between measurement and expected measurement.

An advanced gating network has been developed for interplanetary orbit determination [9]. The structure of the filter bank proposed is illustrated in Figure (2.13). The filter selection in this thesis is not using that type of gating network for filter selection. A similar structure of the filter bank is used in this work. In oder to adapt the operational filter settings a selection scheme making use of the SPKF properties is chosen. The selection logic is based on the difference between the predicted measurements and the actual IMU input.

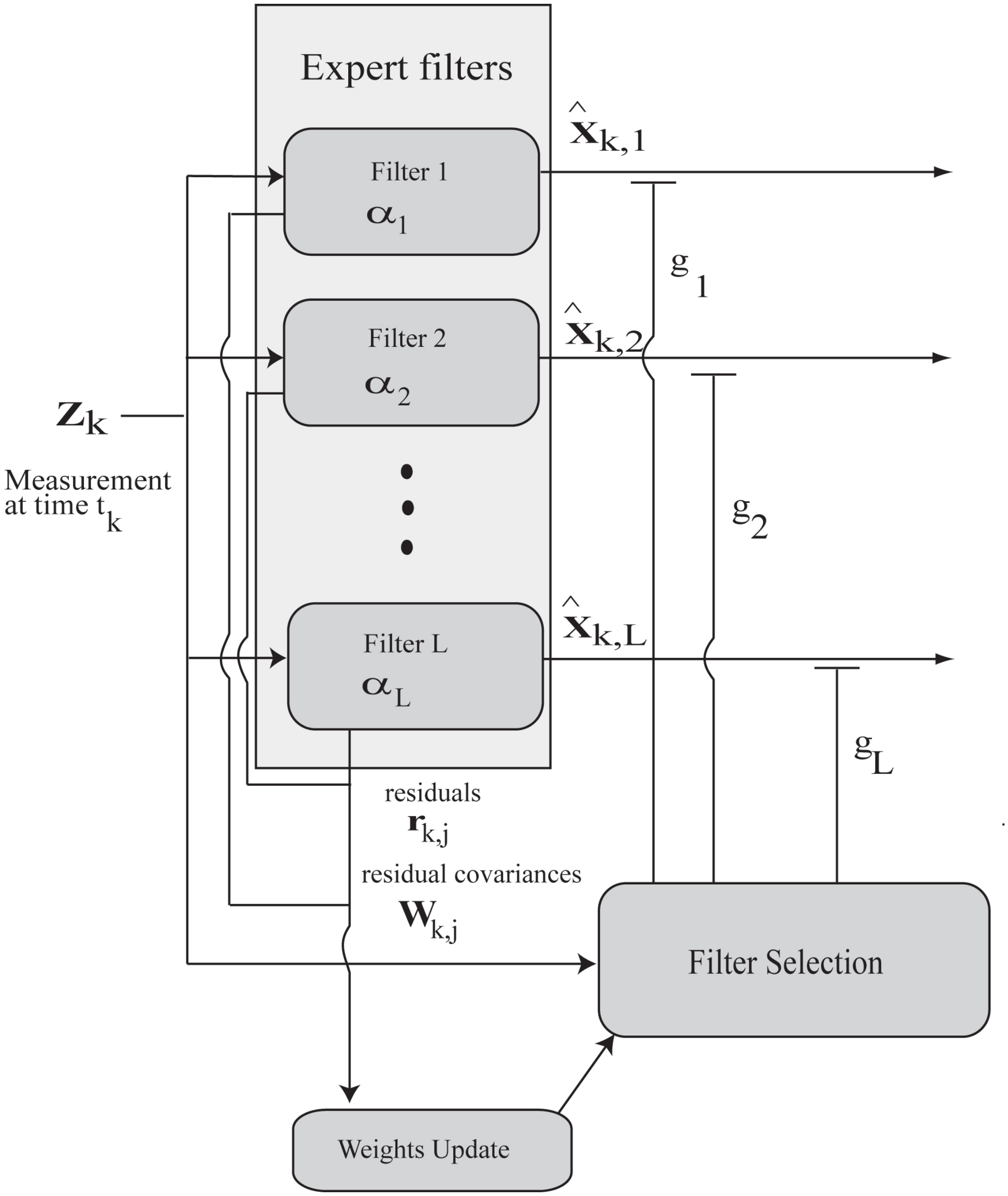

Figure 2.13: Structure of the Filter Bank.

2.6.3 Filter selection

At each time step the performance of all filters in the filter bank is compared and the best performing filter is selected for the next time update. The performance index used to evaluate the performance of each filter is the sum of differences between the predicted measurement vector of each filter and the actual IMU measurement.

$$J = \|\hat{\mathbf{y}}_k - \mathbf{y}_k\| \tag{2.58}$$

The state and covariance of the operational filter are maintained for the next time step. The filter bank parameters of the operational filter are updated from the best performing filter. Then the state and covariance of all filters are reinitialized with the state and covariance of the operational filter for the next time step. Other possible selection schemes are discussed in [9].

2.6.4 Concept Validation

This concept of autonomous environmental parameter identification is capable of determining the encountered environment. As an example for the validity of this approach the filter bank performance in the presence of low noise, Fig. (2.14) shows the filter selected at each time step.

The filter bank contains 11 different atmospheric models. The atmospheric model that is actually encountered during EDL is found in filter 6. The filter models contain different atmospheric densities ρ. Filters 1 through

5 have a less dense atmosphere while filters 6 through 11 contain a denser atmospheric model compared to the actual atmosphere. Filter 0 is the operational filter which adopts the model parameters from the best matching model at each time step.

It can be seen that before the atmosphere has a notable effect on the spacecraft and noise is the major contribution to the measurement, the filter bank selects a parameter set to adjust to the noise of the measurement. Once the atmosphere begins to have a notable effect on the spacecraft at about 50 *sec* into EDL, the filter bank has already converged on the correct atmospheric model. A more detailed analysis of this problem can be found in the discussion of the autonomous parameter identification potential for this filter bank in Chapter 5.

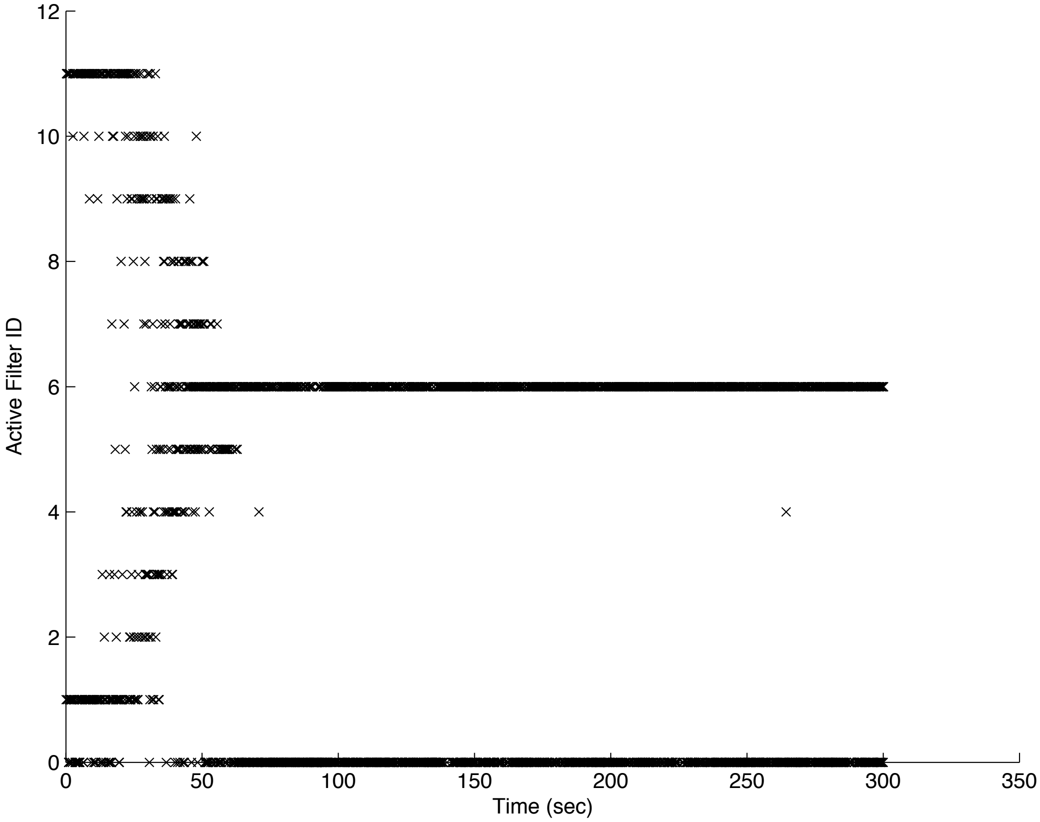

Figure 2.14: Selection of Atmospheric Model in Low Noise.

Chapter 3

Planetary Entry Challenges

3.1 Hypersonic Entry

This work centers around the hypersonic entry phase for a spacecraft at Mars. For successfully completing a pinpoint precision landing accuracy it is essential that a reliable knowledge of the state is available during this mission segment. To date no guided entry at Mars has been performed. During the phase of hypersonic entry the problem preventing accurate spacecraft navigation is founded in the physical constraints encountered during this phase.

During hypersonic EDL the only available method to navigate the spacecraft is dead-reckoning of the IMU data. With initial state uncertainty and noise affecting the IMU, reliable navigation cannot be performed. It is desirable to make adjustments to the spacecraft trajectory at the time when significant aerodynamic forces are encountered at high altitude. Currently the spacecraft is effectively blind as the heat shield obstructs the sensors during this part of the descent. Only the IMU is available during this phase to provide information to the navigation system. This is the problem addressed in this work. At the point where the heat shield is jettisoned and sensors become available to more accurately determine the state of the spacecraft, little time is available to make adjustments to fly the vehicle to a specific location on the surface of the planet.

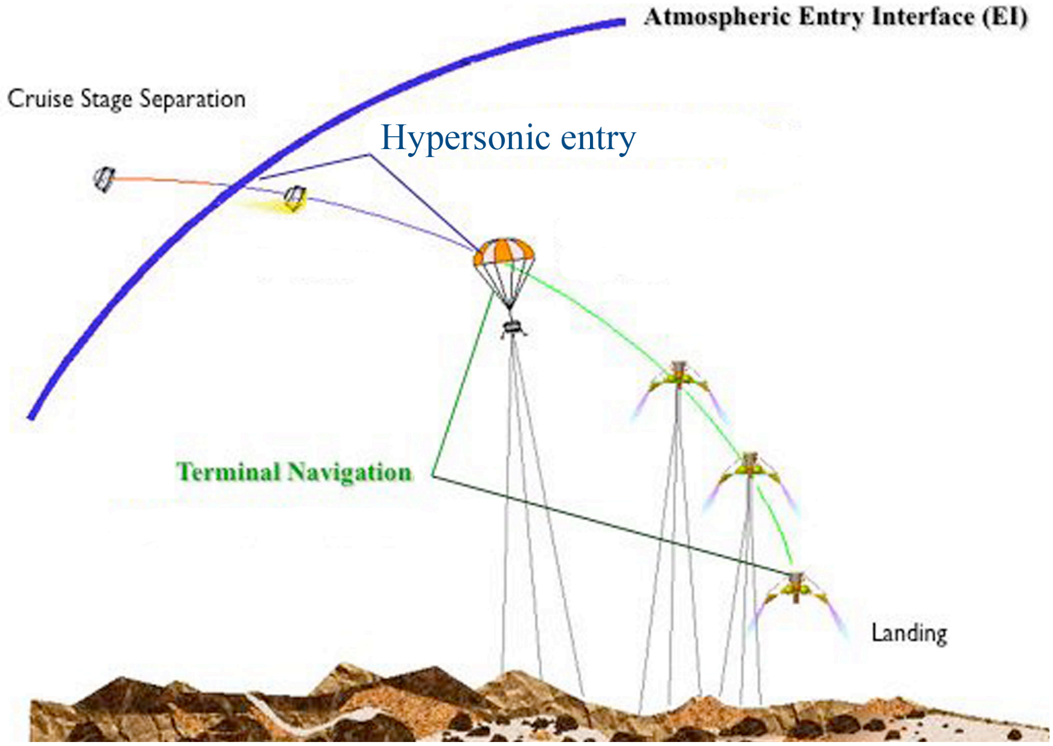

Figure 3.1: Typical Entry, Descent and Landing Scenario, adopted from NASA.

An illustration of a typical Mars EDL is shown in Fig. (3.1). The essential elements relevant for pinpoint precision navigation are depicted in Fig. (3.1). During the short time from cruise stage separation to landing on the planet, there is the hypersonic entry phase extending from entry interface (EI) to parachute deploy where there is significant ability to control the trajectory using aerodynamic lift. For the nominal MER mission the hypersonic phase was about 242 seconds long. During this time, the vehicle passes through peak heating and peak aerodynamic forces.

For successful pinpoint precision navigation, it is required to have accurate navigation capability during the high-altitude hypersonic entry segment to take advantage of the aerodynamic forces present and guide it to the desired touchdown location. By using the IMU measurements as external measurements in a model-based navigation filter architecture, the spacecraft state can be estimated to the accuracy required to execute guidance concepts and successfully maneuver the spacecraft to the target. The goal of autonomy is addressed by using a bank of sigma point Kalman filters.

For an entry at Mars, atmospheric uncertainty is the grand challenge for autonomous spacecraft entry. The atmosphere of Mars fluctuates significantly and even high-grade atmospheric models like the 2005 Mars GRAM are only moderately capable to predict the environment accurately. For a real-time navigation system less computationally intensive models need to be used increasing the uncertainty in the environment. It is imperative to accurate navigation that the state of the atmosphere be determined properly. This can be addressed through a bank of sigma point Kalman filters for environmental parameter identification.

The proposed concept is verified using MER mission data in simulation and hosted on the Electra programmable radio as a navigation platform. A MER-like entry from MSL cruise stage separation conditions is considered as a second case illustrating the validity of the proposed navigation architecture for a different scenario.

3.2 Mars Exploration Rover Mission

The Rover A (Spirit) mission used a standard Delta II 7925 when it launched June 10, 2003. The later Rover B (Opportunity) launch on July 7, 2003, needed more energy to get to Mars, so it launched on a Delta II 7925 Heavy. The spacecraft design for the Mars Exploration Rover mission is largely based on the successful Mars Pathfinder system for entry, descent, and landing. The rover design is based on the Athena Rover on the previously cancelled Mars 2001 lander mission.

During the cruise phase the spacecraft was tracked from earth through the use of NASA Deep Space Network (DSN), which is an international network of antennas that provide the tracking and communication links between the scientists and engineers on Earth and the Mars Exploration Rovers in space and on Mars. The DSN consists of three deep-space communications facilities placed approximately 120 degrees apart around the world: at Goldstone, in the California Mojave Desert; near Madrid, Spain; and near Canberra, Australia. This strategic placement permits constant observation of spacecraft as the Earth rotates on its own axis.

The entry, descent, and landing (EDL) phase begins when the spacecraft reaches the Mars atmospheric entry interface point (3522.2 kilometers or about 2,113 miles from the center of Mars or 125 kilometers above the surface) and ends with the lander on the surface of Mars in a safe state.

Spirit landed in Gusev crater on January 4, 2004 at 04:35 Ground UTC during the latter half of the northern winter/southern summer on Mars. Opportunity landed in the Meridiani Planum on the opposite side of Mars from Spirit, on January 25, 2004 05:05 Ground UTC. That means that both rovers landed in the Martian afternoon while the Earth is still in view, allowing the Earth to receive the landing signal if the lander is on the base petal. The rovers are still in operation 1000 martian days past their expected end of life.

3.2.1 Entry Sequence Profile

Starting at EI, MER followed a typical EDL sequence until reaching the surface of the planet. As demonstrated by the Pathfinder mission, the final landing method utilized the concept of airbag landing to protect the craft at touchdown.

Shortly after EI, the atmosphere of Mars began to have a measurable effect. This led to the portions of high dynamic and thermal loads on the spacecraft. After passing though this most crucial hypersonic phase of the entry, the hypersonic drag chute was deployed and the heat shield disposed. At an appropriate point, the subsonic parachute was deployed and the spacecraft descended to the surface of the planet. Shortly before impact the airbags were deployed and the spacecraft gradually rolled to a halt and could then release the rover.

3.3 Mars Science Laboratory Mission

Building on the success of the two rover geologists that arrived at Mars in January, 2004, NASA's next rover mission is being planned for travel to Mars before the end of the decade. Twice as long and three times as heavy as the Mars Exploration Rovers Spirit and Opportunity, the Mars Science Laboratory will collect martian soil samples and rock cores and analyze them for organic compounds and environmental conditions that could have supported microbial life now or in the past. The mission is anticipated to have a truly international flavor, with a neutron-based hydrogen detector for locating water provided by the Russian Federal Space Agency, a meteorological package provided by the Spanish Ministry of Education and Science, and a spectrometer provided by the Canadian Space Agency.

Mars Science Laboratory is intended to be the first planetary mission to use precision landing techniques, steering itself toward the martian surface similar to the way the space shuttle controls its entry through the Earths upper atmosphere. In this way, the spacecraft would fly to a desired location above the surface of Mars before deploying its parachute for the final landing. As currently envisioned, in the final minutes before touchdown, the spacecraft would activate its parachute and retro rockets before lowering the rover package to the surface on a tether (similar to the way a skycrane helicopter moves a large object). This landing method would enable the rover to land in an area 20 to 40 kilometers (12 to 24 miles) long, about the size of a small crater or wide canyon and three to five times smaller than previous landing zones on

Mars.

Like the twin rovers now on the surface of Mars, Mars Science Laboratory would have six wheels and cameras mounted on a mast. Unlike the twin rovers, it would carry a laser for vaporizing a thin layer from the surface of a rock and analyzing the elemental composition of the underlying materials. It would then be able to collect and crush rock and soil samples and distribute them to on-board test chambers for chemical analysis. Its design includes a suite of scientific instruments for identifying organic compounds such as proteins, amino acids, and other acids and bases that attach themselves to carbon backbones and are essential to life as we know it. It could also identify features such as atmospheric gases that may be associated with biological activity.

Using these tools, Mars Science Laboratory would examine martian rocks and soils in greater detail than ever before to determine the geologic processes that formed them; study the martian atmosphere; and determine the distribution and circulation of water and carbon dioxide, whether frozen, liquid, or gaseous. NASA plans to select a landing site on the basis of highly detailed images sent to Earth by the Mars Reconnaissance Orbiter beginning in 2006, in addition to data from earlier missions.

NASA is considering nuclear energy for powering the Mars Science Laboratory. The rover would carry a U.S. Department of Energy radioisotope power supply that would generate electricity from the heat of plutonium's radioactive decay. This type of power supply could give the mission an operating lifespan on Mars' surface of a full martian year (687 Earth days) or more. NASA is also considering solar power alternatives that could meet the mission's science and mobility objectives.

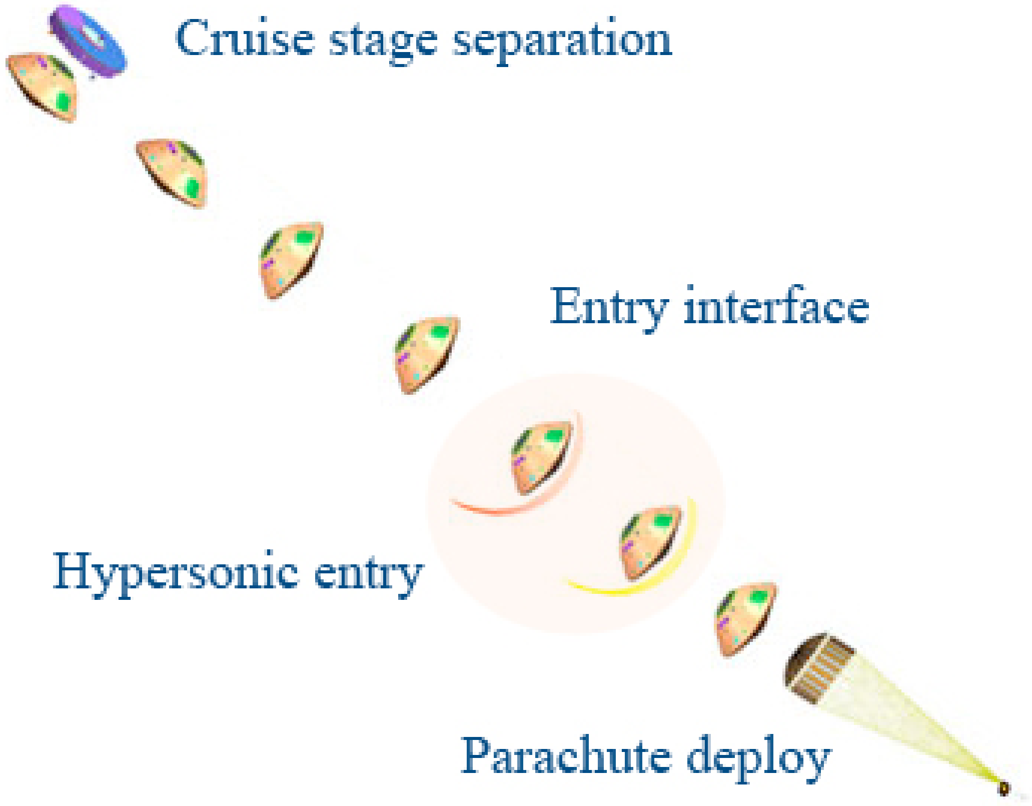

Figure 3.2: MSL Hypersonic Entry Phase to Parachute Deploy, adopted from NASA.

3.3.1 Mission Profile

Plans for the Mars Science Laboratory call for launch from Florida, in September or October 2009 and arrival at Mars in summer 2010. The spacecraft is being designed to steer itself during descent through Mars' atmosphere with a series of S-curve maneuvers similar to those used by astronauts piloting NASA space shuttles. During the 3 minutes before touchdown the spacecraft would slow its descent with a parachute, then use retro rockets mounted around the rim of an upper stage for the final 500 meters (1,640 feet) of the descent. In the final seconds, the hovering upper stage would act as a sky crane, lowering the upright rover on a tether to the surface. As envisioned, the mobile laboratory itself will be about twice as long (about 2.8 meters or 9 feet) and four times as heavy as NASA's twin Mars Exploration Rovers launched in 2003. It would inherit some design elements from them, including six-wheel drive, a rocker-bogie suspension system and cameras mounted on a mast to help the mission's Earthbound humans select exploration targets and driving routes. Unlike earlier rovers, Mars Science Laboratory is being designed to carry equipment to gather samples of rocks and soil, crush them and distribute them to onboard test chambers inside analytical instruments. NASA's Jet Propulsion Laboratory, Pasadena, Calif., builder of the Mars Science Laboratory, is engineering the rover to roll over obstacles up to 65 centimeters (25 inches) high and to travel up to about 200 meters (660 feet) per day on martian terrain. The mission is being designed to use radio relays via Mars orbiters as the principal means of communication between the Mars Science Laboratory and Earth. The EDL phase is illustrated in Fig (3.2). After cruise stage separation the spacecraft will perform a series of maneuvers to turn to the entry attitude. After passing entry interface the spacecraft will pass the peak thermal and aerodynamic loads and then turn to parachute deploy heading and conclude the hypersonic entry by deploying the hypersonic parachute.

3.3.2 Mission Entry Profile

As MSL will be used to demonstrate autonomous precision landing at Mars, a significant difference to the Pathfinder and MER mission will be the last step of EDL. Instead of relying on airbags to safely place the spacecraft on the surface of the planet, MSL will use a new concept to place the rover in it's desired location.

As the exact spot for the lander cannot be picked when using airbags due to the circumstance that the craft will roll to a gradual halt, MSL will set down the payload rover in an exact spot, picked in the landing site acquisition phase of the EDL segment.

The post-hypersonic final EDL begins by slowing the spacecraft with a parachute and jettison of the heat shield. The radar then has new sensors available for the final corrections and landing site acquisition. After release of the backshell with the parachute a powered descent flies the spacecraft to the required altitude to use a unique concept during which MSL will make use of the concept of a sky crane helicopter to place the rover on the surface. Upon picking a landing site, the craft will descend to an altitude of 10 feet and lover the rover on a tether to the surface of the planet. After the rover is successfully placed on the surface of the planet, the craft's structure will fly off and crash away from the rover.

MSL is a potential mission for application of the navigation concept proposed in this dissertation. Particularly for the phase of hypersonic entry, MSL has the exact constraints and requirements which are addressed. It is necessary to achieve pinpoint accuracy for arriving at the touchdown site desired for the rover. Therefore the spacecraft will have to adjust the trajectory in the hypersonic phase when only IMU data is available. Processing this data as an external measurement in a model-based bank of sigma point kalman filters will provide accurate spacecraft state knowledge and enable it to make autonomous adjustments.

Chapter 4

Model Uncertainties

This chapter describes the computational implementation of the entry simulations and SPKF models. The simulations are performed for the purpose of analyzing the proposed navigation architecture.

The first scenario is based on the MER-A entry at Mars. Beginning at entry interface a state and covariance are available [25]. Taking this as the starting condition, an entry trajectory was simulated for a non-lifting, spinning spacecraft up to hypersonic parachute deployment conditions. This trajectory was compared to a JPL reconstructed trajectory for MER-A and matched closely.

With the entry trajectory, IMU measurements were created. The noise free set of IMU measurements combined with the initial state were used to create a reconstructed trajectory which matched the reference trajectory. The IMU measurements were then corrupted with noise of different levels. These noisy measurements were used to generate a dead-reckoned trajectory in the presence of low, medium and high noise to be used in the analysis of the SPKF performance.

A second scenario is considered which begins at cruise stage separation conditions of a proposed MSL entry. The analysis was again done for a non-lifting, spinning spacecraft. Again a simulated trajectory was created, IMU measurements recorded for this reference and these used for reconstruction and dead-reckoning.

4.1 Dynamic Models

The state vector consists of position, $\mathbf{r}$, velocity, $\mathbf{v}$ and the attitude in form of a quaternion, $\mathbf{Q}$, or

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \\ \mathbf{Q} \end{bmatrix} \tag{4.1}$$

The dynamic equations implemented in the filter represent the spacecraft equations of motion during EDL. It is formulated as follows:

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{Q}} \end{pmatrix} = \begin{pmatrix} \mathbf{v} \\ \mathbf{a}_{gravity} + \mathbf{a}_{drag} \\ \frac{1}{2}\mathbf{A}(\omega)\mathbf{Q} \end{pmatrix} \tag{4.2}$$

where

$$\mathbf{A}(\omega) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix}. \tag{4.3}$$

The angular velocity ω is obtained from the IMU data of gyro differences in angles $\omega = \Delta\theta/\Delta t$. For the simulation and filter implementation the angular velocity is assumed to be a constant rotation about the velocity vector, or

$$\omega = \omega_0 \; \mathbf{e}_{v_r}. \tag{4.4}$$

where

$$\mathbf{v}_r = \mathbf{v} - \omega_{mars} \times \mathbf{r} \quad \text{and} \quad \mathbf{e}_{v_r} = \frac{\mathbf{v}_r}{\|\mathbf{v}_r\|} \tag{4.5}$$

The spin rate ω_0 is a constant. This is an approximation of the actual spin rate which varied. The accelerations considered in the spacecraft dynamics are gravity, given by

$$\mathbf{a}_{gravity} = -\frac{\mu}{||\mathbf{r}||^3}\ \mathbf{r} \tag{4.6}$$

and atmospheric drag, given by

$$\mathbf{a}_{drag} = -\frac{1}{2}\frac{\rho C_D A||\mathbf{v}_r||^2}{m}\mathbf{e}_{v_r}. \tag{4.7}$$

These dynamics are used to simulate the spacecraft trajectory from EI to parachute deploy. The gravitational parameter μ is assumed known for Mars. The remaining parameters C_D, A, and m are also assumed known, although uncertainty (especially in C_D) will lead to navigation errors. However, since C_D appears in Eq. (4.7) and we are adapting to different models for ρ, then variation in C_D will be captured.

4.2 Environmental Uncertainties

The environment presents several areas of uncertainty. Gravity and atmosphere can be modeled in different ways, and each may require knowledge of various parameters. The main uncertainty for the Mars EDL problem is the atmospheric density, but other uncertainties, especially in the gravity field, do exist.

Multiple atmospheric models exist for Mars. The one developed in this work is a multi-layer exponential atmosphere that was constructed using Mars-Gram 2005. The idea is to compute a reference density ρ_0 for each layer with a fixed scale height, h_0 and to fit the exponential model given by

$$\rho = \rho_0 \exp^{-\frac{h}{h_0}} . \tag{4.8}$$

The altitude h is measured above the planet surface. The atmospheric density is one of several variables that are needed to determine drag forces acting upon the spacecraft during the atmospheric flight segment. For the purpose of precision navigation, a high-quality model must be available for the specific entry time, landing site and entry trajectory. A multi-layer model of the atmosphere was constructed using 20 layers. The atmospheric model has a significant influence on the spacecraft entry navigation.

4.2.1 The Mars Multilayer Atmosphere

The martian atmosphere is modeled as a multilayer exponential atmosphere. The modeling is accomplished by performing a least square fit using Mars Gram 2005 as the source of the density. Beginning with the density function

$$y = c \exp^{bx} = \exp^{a} \exp^{bx} \tag{4.9}$$

we take the natural logarithm to obtain

$$\ln y = a + bx, \tag{4.10}$$

and define

$$z = a + bx. \tag{4.11}$$

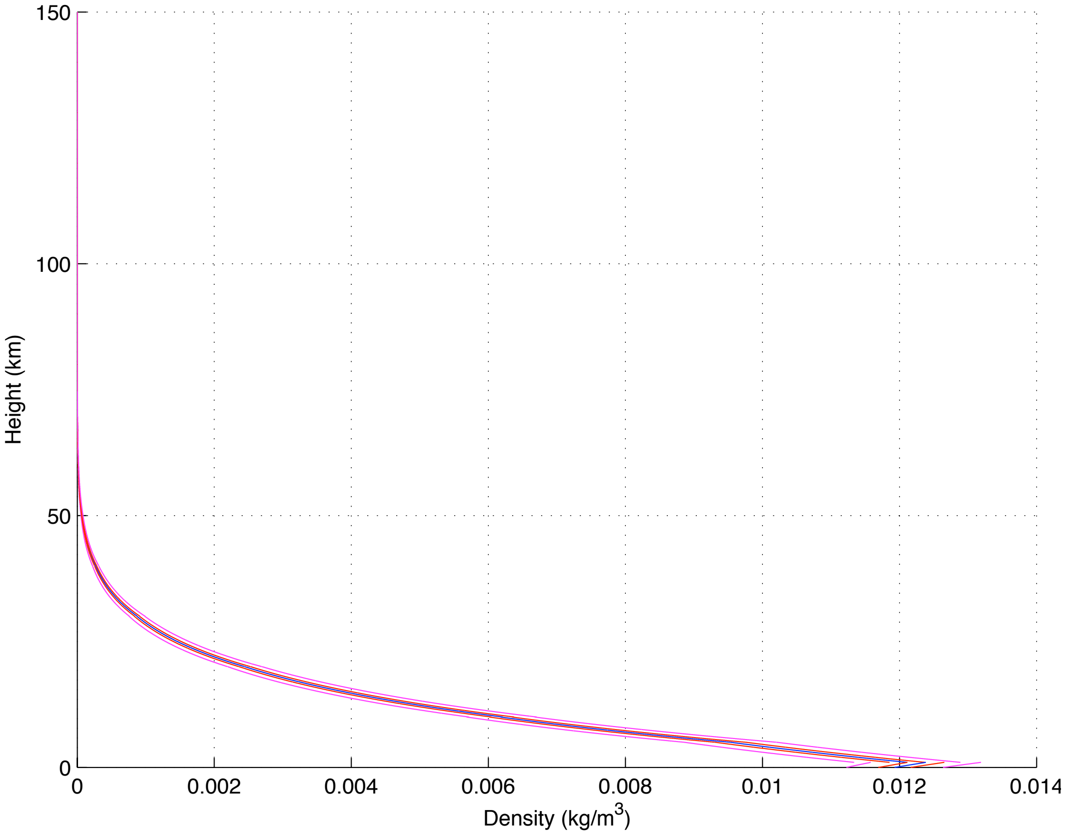

Figure 4.1: Mars Multi Layer Atmospheric Model 0 km to 150 km.

Here x is the altitude and z the natural logarithm of the density. The density is obtained by calling Mars Gram 2005 for the specific landing site and altitude. This process is done for the nominal atmosphere and $\pm\sigma$ and $\pm 3\sigma$ atmospheres. For a and b this results in

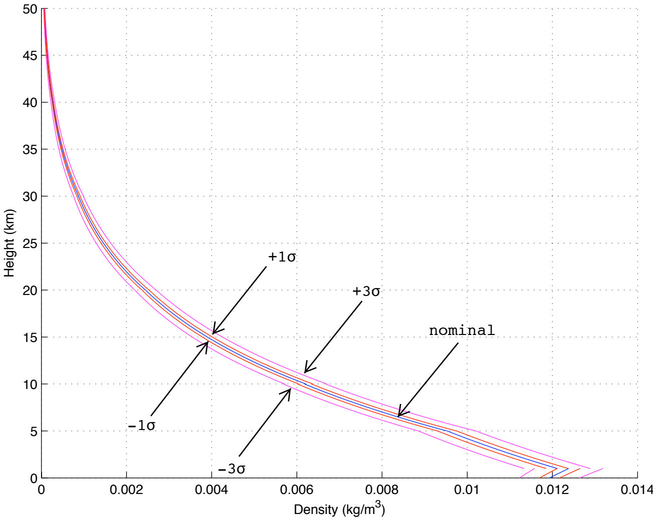

Figure 4.2: Mars Multi Layer Atmospheric Model 0 km to 50 km.

$$a = \frac{\sum_{i=1}^{n} z \sum_{i=1}^{n} x^2 - \sum_{i=1}^{n} x \sum_{i=1}^{n} xz}{n \sum_{i=1}^{n} x^2 - (\sum_{i=1}^{n} x)^2} \tag{4.12}$$

and

$$b = \frac{n \sum_{i=1}^{n} xz - \sum_{i=1}^{n} x \sum_{i=1}^{n} z}{n \sum_{i=1}^{n} x^2 - (\sum_{i=1}^{n} x)^2} \tag{4.13}$$

with n being the number of reference points for the atmospheric layer considered. The multi-layer atmosphere is shown in Fig. (4.1) through Fig. (4.4).

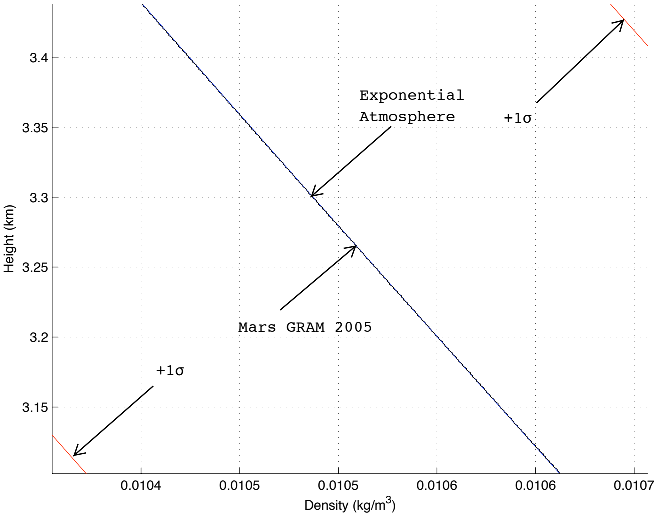

Figure 4.3: Mars Multi Layer Atmosphere 1 Sigma Range.

4.3 Measurement Models

4.3.1 Actual Measurements

The data obtained from the IMU are processed during the simulation to update the state at a nominal time interval assumed to be $\frac{1}{8}$ of a second. The measurements provided are based on differences in inertial velocity of the spacecraft from the previous measurement to the current time. These $\Delta\mathbf{v}$ are adjusted for the effect of gravity and transformed into the IMU frame.

Noise is taken into account in two different parts of the analysis. The

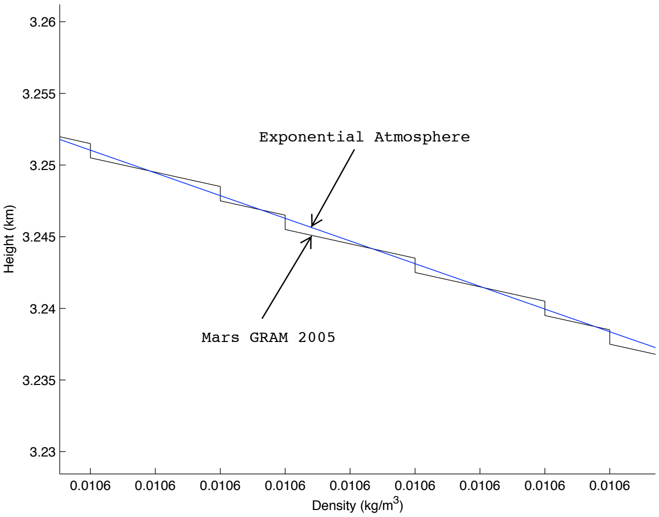

Figure 4.4: Mars Multi Layer Atmosphere Difference from Mars GRAM 2005.

measurements obtained from the IMU are perturbed by noise. Also, uncertainties in the dynamic model are taken into account in the form of process noise.

4.3.2 Filter Measurement Model

As the attitude is dead-reckoned (not filtered with the SPKF), the only measurement that needs to be modeled is $\Delta \mathbf{v}$. This is measured in the IMU coordinate frame and modeled accordingly. The non-gravitational effect is the acceleration due to drag and is the modeled $\Delta \mathbf{v}$ and is calculated in the filter as

$$\Delta \mathbf{v}_{modeled} = \Delta \mathbf{v}_{drag,IMU} + \Delta \mathbf{v}_{offset}. \tag{4.14}$$

To account for the effect of the spacecraft rotation measured as part of the IMU measurement, the term

$$\Delta \mathbf{v}_{offset} = \mathbf{T}_{SC \rightarrow IMU} (\omega \times \omega \times \mathbf{r}_{offset}) \Delta t \tag{4.15}$$

with the known position vector of the IMU with respect to the spacecraft center of mass $\mathbf{r}_{offset}$.

The $\Delta \mathbf{v}$ due to drag is computed by

$$\Delta \mathbf{v}_{drag,IMU} = \mathbf{T}_{inertial \rightarrow IMU} \Delta \mathbf{v}_{drag,inertial}. \tag{4.16}$$

where the transformation $\mathbf{T}_{inertial \rightarrow IMU}$ is function of the estimated attitude, $\hat{\mathbf{Q}}$. The term $\Delta \mathbf{v}_{drag,inertial}$ is given by

$$\Delta \mathbf{v}_{drag,inertial} = (\mathbf{v}_i - \mathbf{v}_{i-1}) - \Delta \mathbf{v}_{gravity} \tag{4.17}$$

4.3.3 IMU Measurements

The IMU measurements are modeled after an MER-class spacecraft. This means that we assume a non-lifting spacecraft spinning at constant rate about the relative velocity vector. The simulated measurements were compared to the MER-A entry data showing a close match. For the entry trajectory, computed IMU measurements are simulated. Three noise levels are considered. The measurements (shown in the IMU frame) are discrete measurements at $8Hz$. The plot for low noise is Fig. (4.5). The plot for medium noise is Fig. (4.6). The plots for high noise are Fig. (4.7) and Fig. (4.8). The level of measurement noise is defined by the noise covariance of the measurements as shown in Table 4.1.

Low	Medium	High
$4e^{-8}$	$4e^{-4}$	$4e^{-2}$

Table 4.1: Noise Covariance Levels $[\frac{m^2}{s^2}]$

The IMU measurements are processed in this thesis without modeling bias, scale errors and misalignments. These error sources were not the focus of this work and will only result in reduced performance of the dead-reckoning process. These errors will manifest themselves as measurement dependent noise. Rather than modeling these error sources they are taken into account in the calculation of the state covariance by adjusting the process noise accordingly. They are not represented in the state however.

A sample run has been performed considering constant bias in the IMU which showed the severe effect on the dead-reckoning process. The model-based continued to perform well in the experiment while the dead-reckoning quickly diverged.

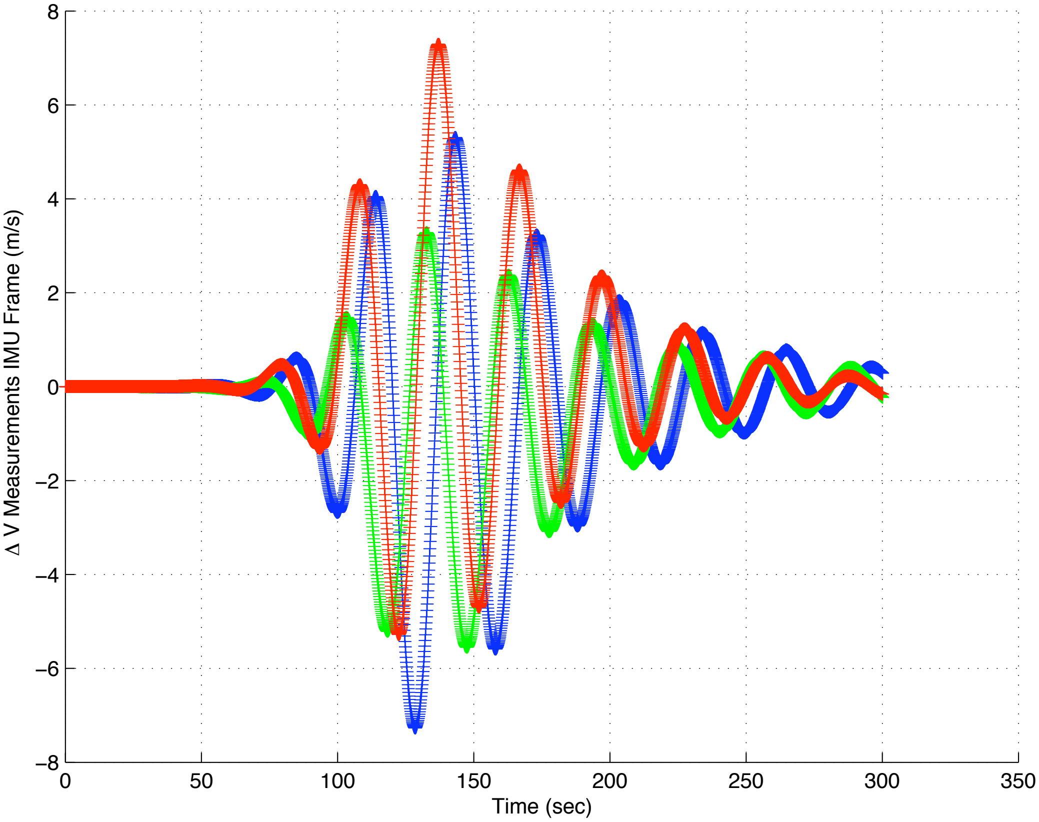

Figure 4.5: Discrete IMU delta V Measurements with Low Noise.

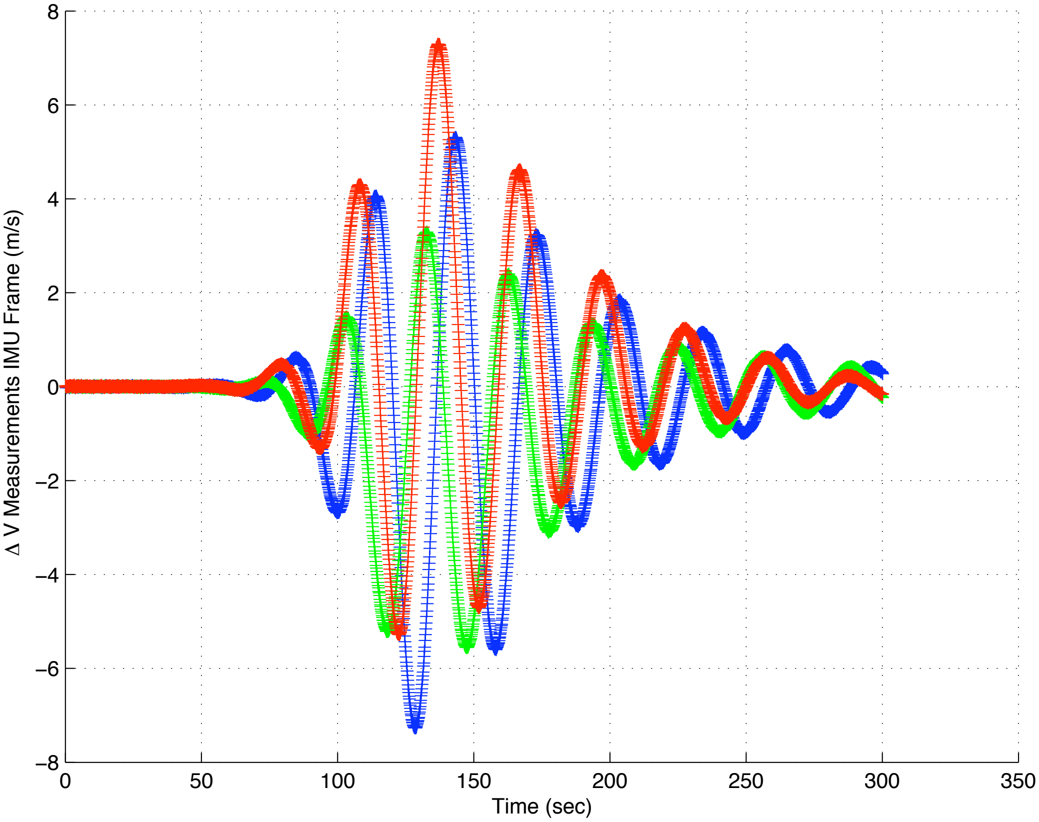

Figure 4.6: Discrete IMU delta V Measurements with Medium Noise.

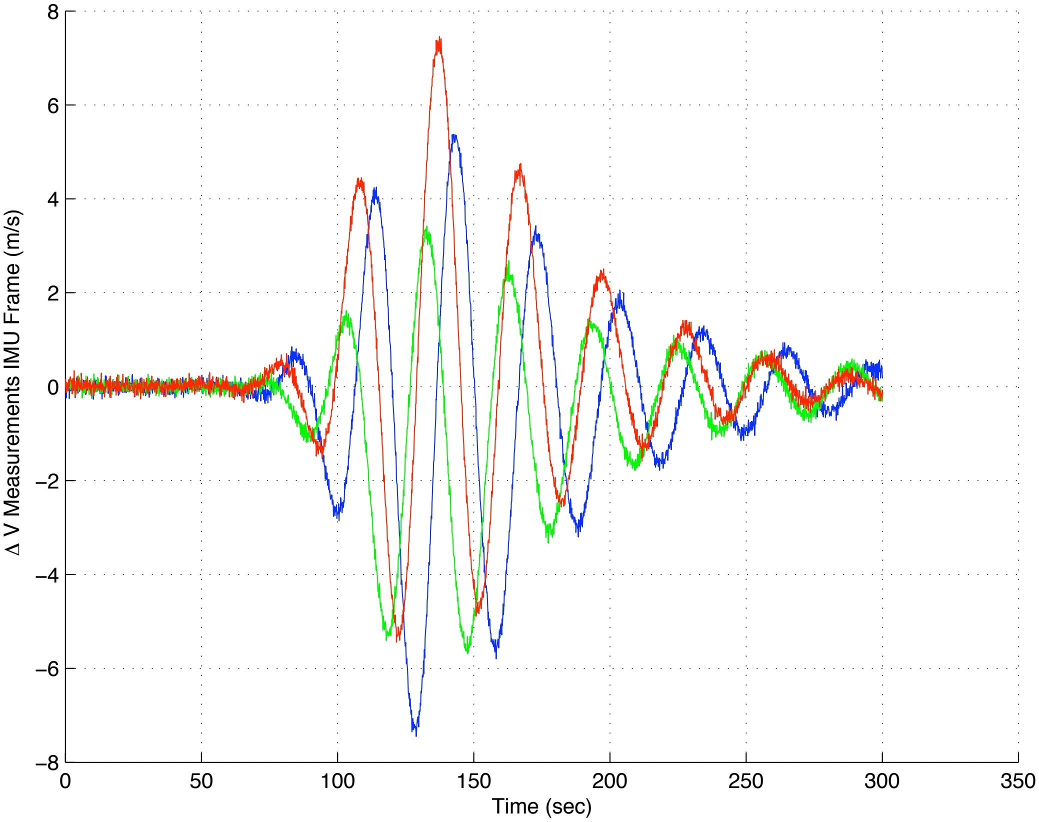

Figure 4.7: Continuous IMU delta V Measurements with High Noise.

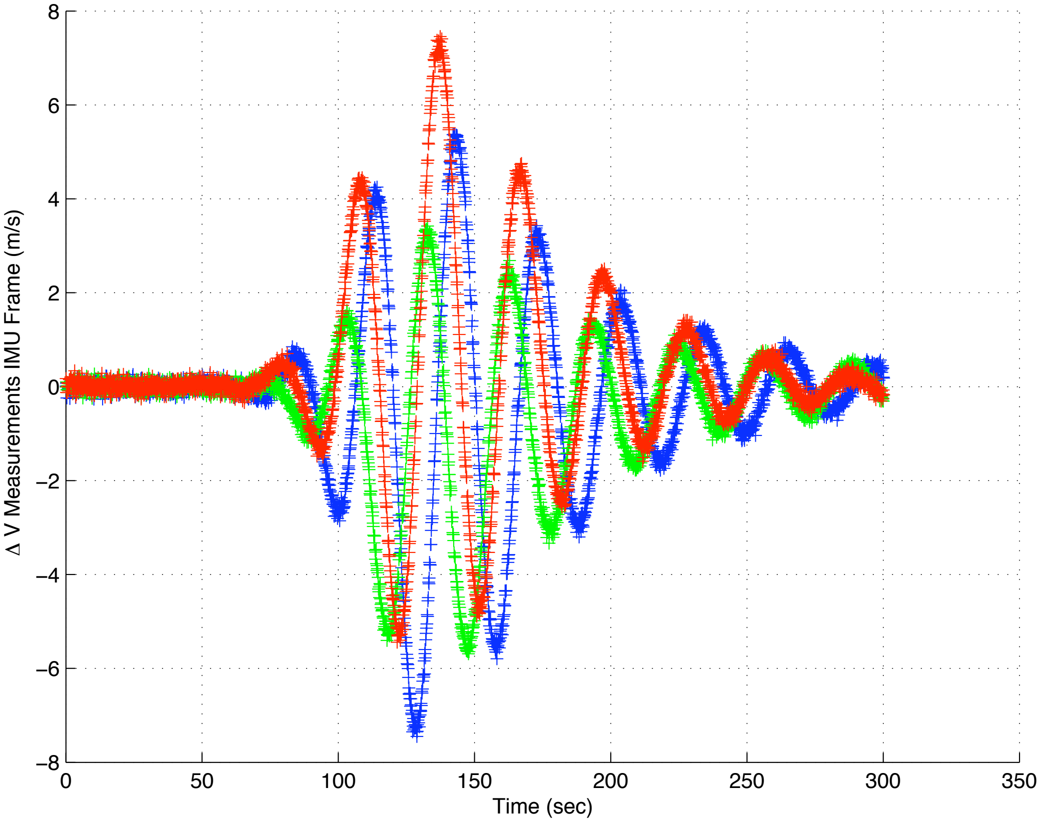

Figure 4.8: Discrete IMU delta V Measurements with High Noise.

4.3.4 IMU Measurements for Second Scenario

The second mission profile is considered, also as a non-lifting spinning spacecraft entry scenario. Beginning at cruise stage separation the spacecraft is considered to be spinning at a constant rate during the EDL phase. For the trajectory computed, IMU measurements are simulated. Two noise levels are considered. The measurements are plotted in the IMU frame and shown as discrete measurements at $8Hz$.

The plot for medium noise is Fig. (4.9). The plot for high noise is Fig. (4.10).

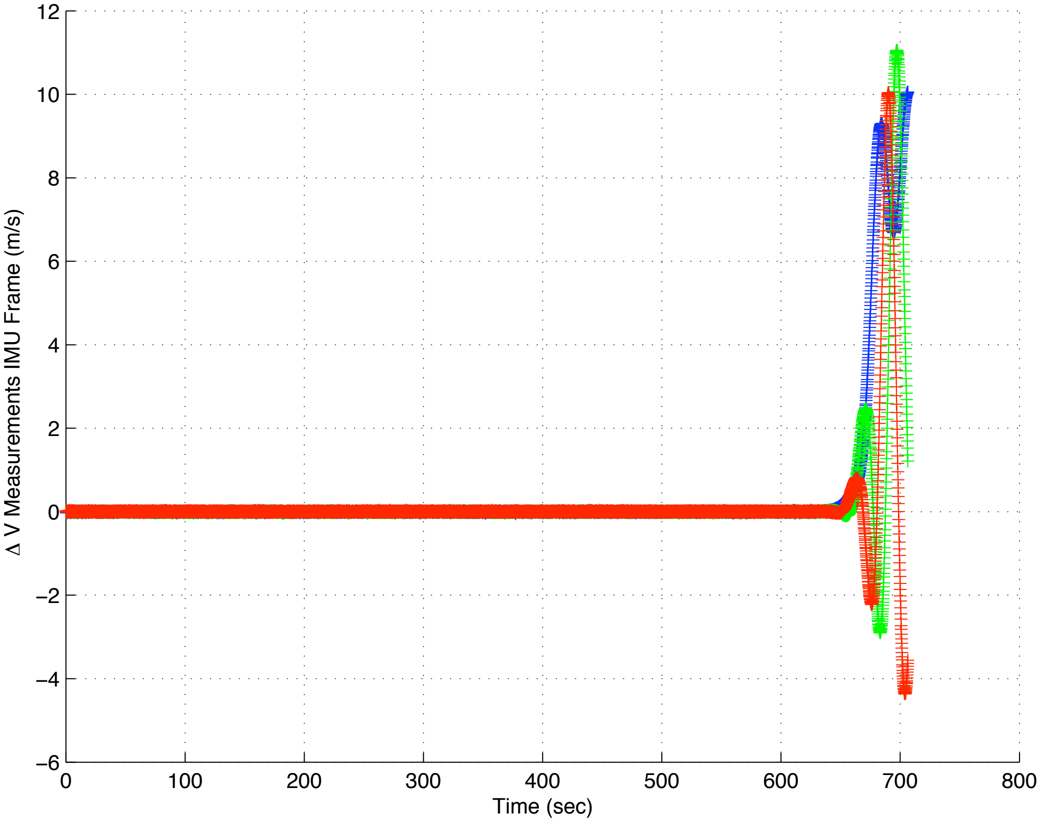

Figure 4.9: Discrete IMU delta V Measurements for Second Scenario with Medium Noise.

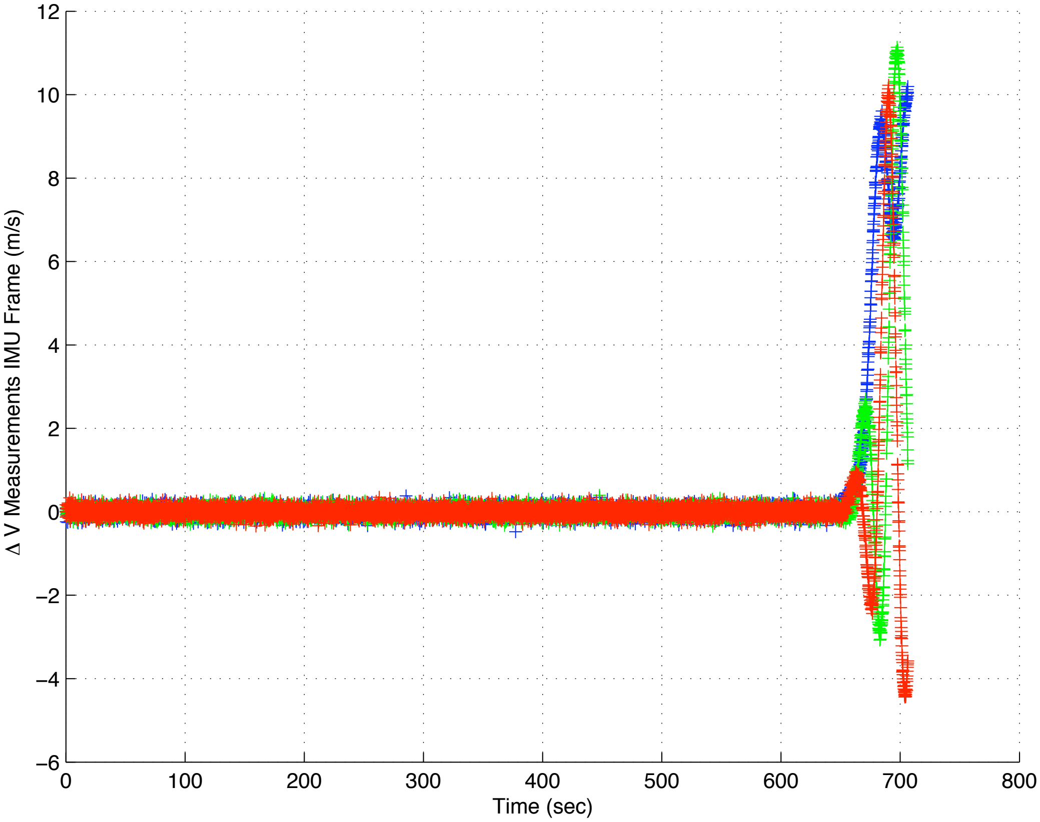

Figure 4.10: Discrete IMU delta V Measurements for Second Scenario with High Noise.

4.4 EDL Trajectories

4.4.1 Trajectory Simulation

In order to test the filter estimates a good entry trajectory must be available. This is obtained by performing a simulation based on EI data provided by JPL. From this point forward the reference trajectory is created through a simulation with known parameter setting for the spacecraft and environment along with a defined dynamic model given to the simulator.

As an entry trajectory for MER-A was provided by JPL in form of a reconstruction from mission data, the simulated trajectory is compared against this data set to validate the models used.

4.4.1.1 First Scenario

With JPL data available an entry trajectory is created for MER. The models used are described in 4.

The UT simulated trajectory is shown in Fig. (4.12) through Fig. (4.21). This trajectory includes position, velocity and attitude of the spacecraft.

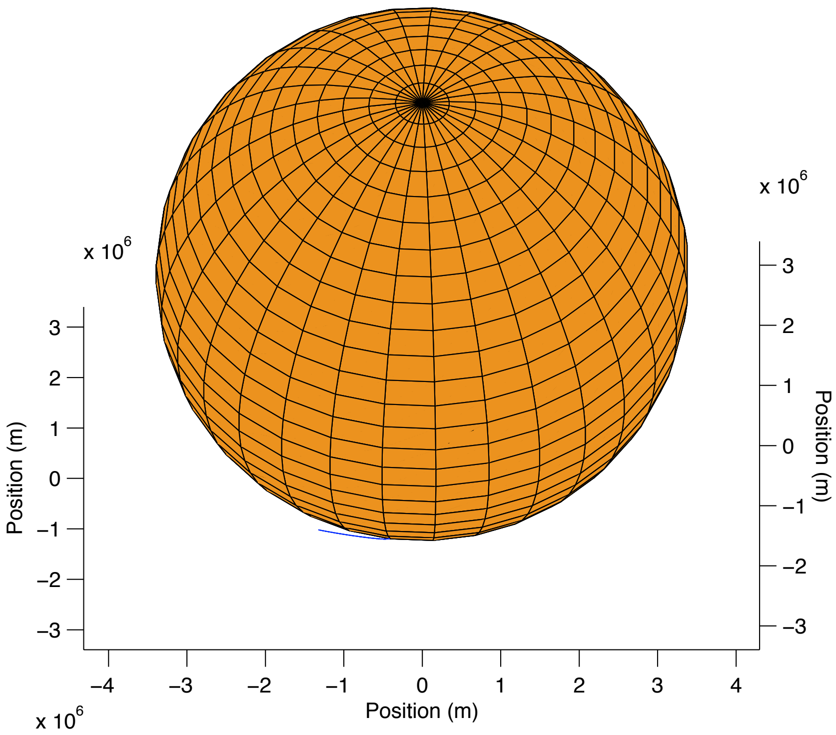

Figure 4.11: Entry Trajectory for First Scenario.

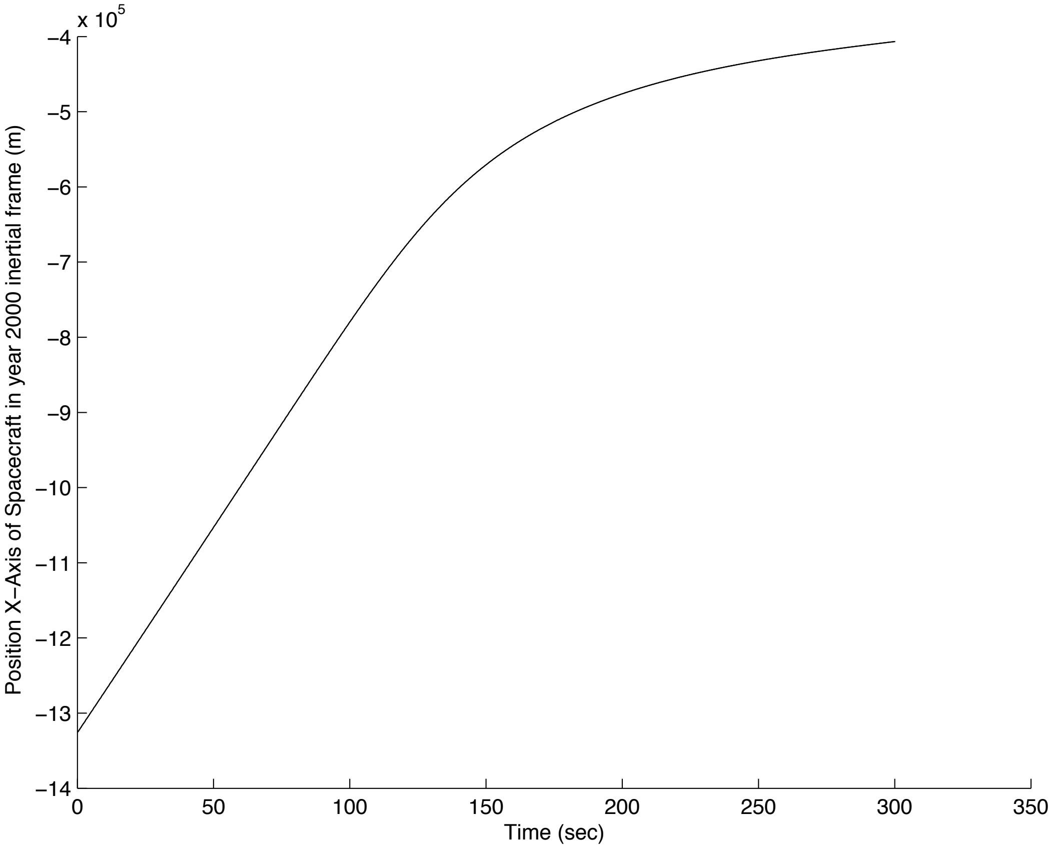

Figure 4.12: Simulated Entry Trajectory for First Scenario year 2000 x.

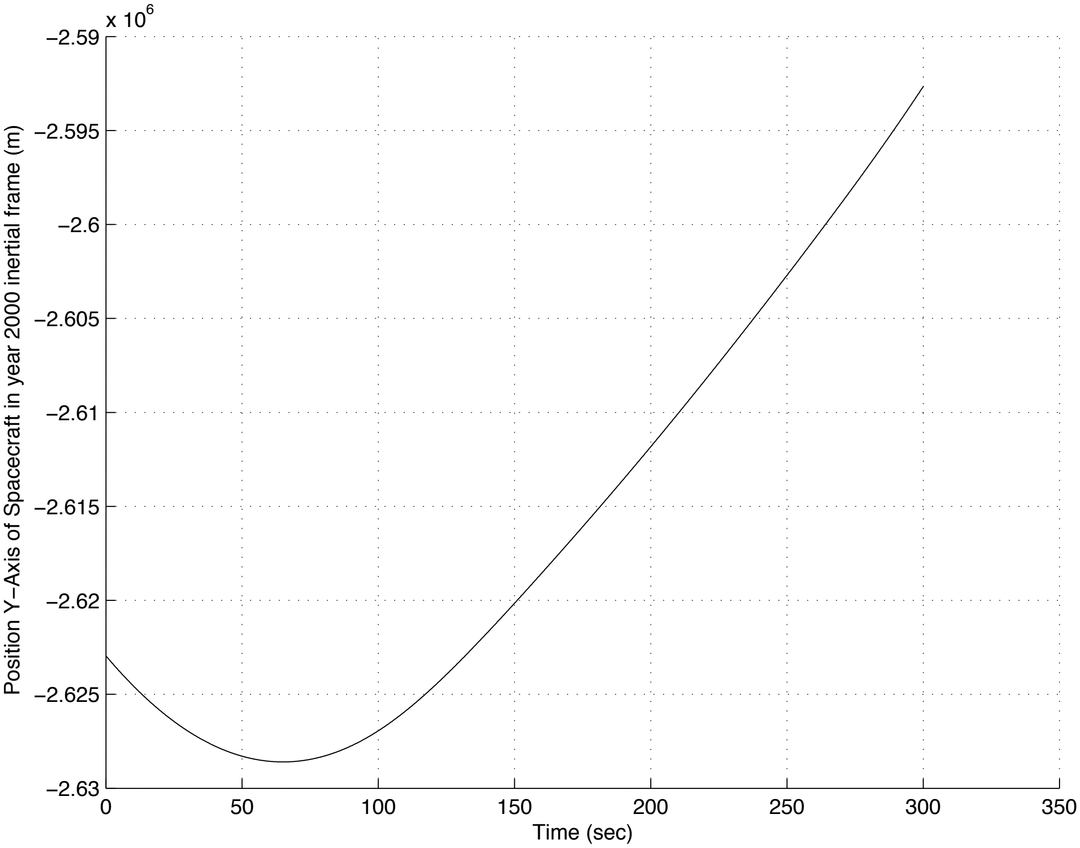

Figure 4.13: Simulated Entry Trajectory for First Scenario year 2000 y.

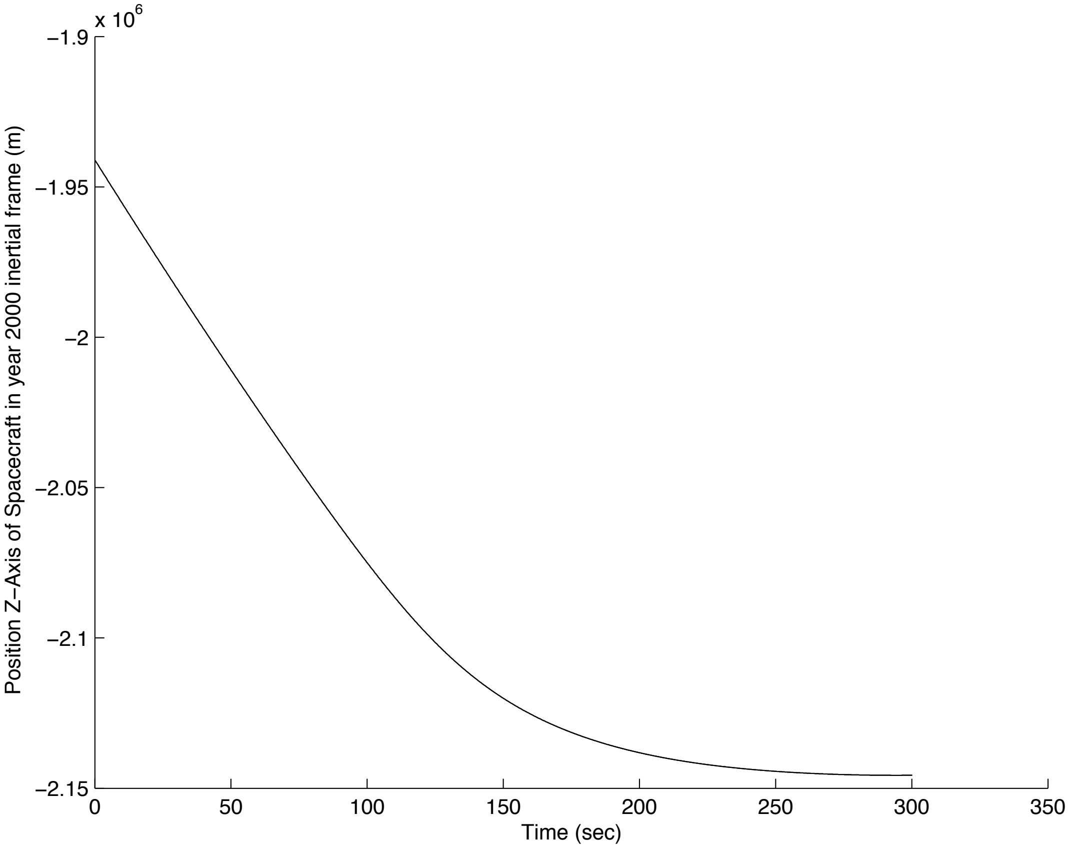

Figure 4.14: Simulated Entry Trajectory for First Scenario year 2000 z.

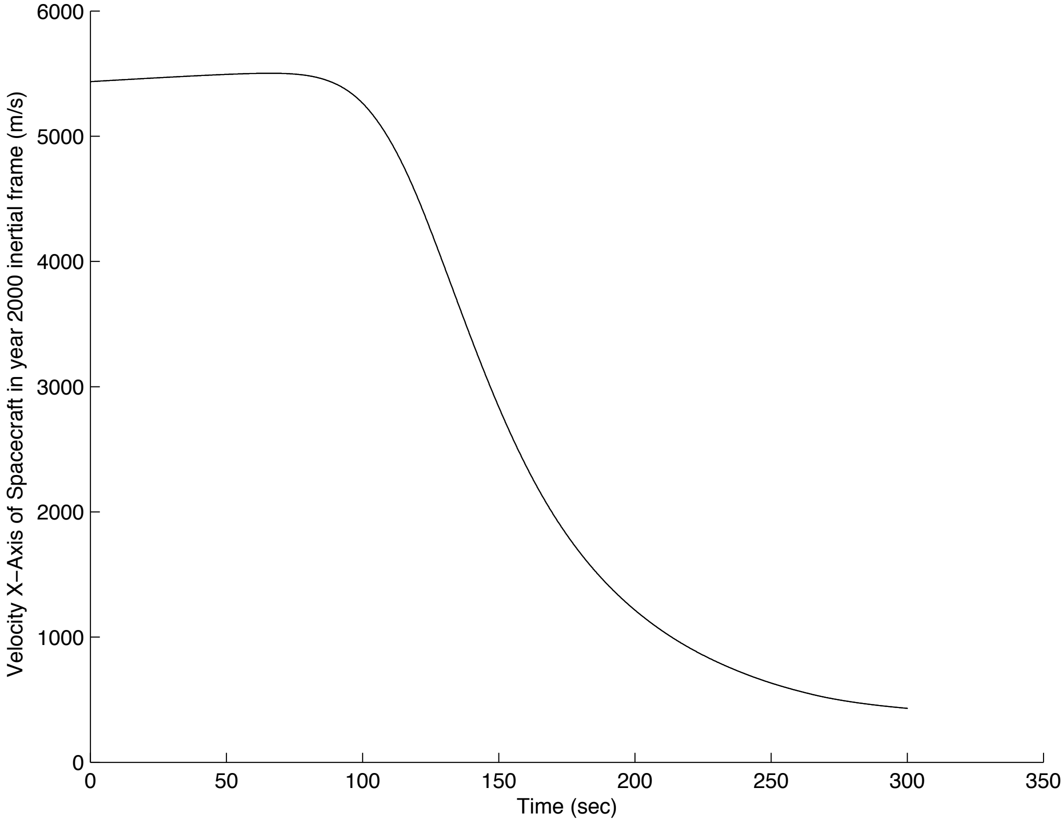

Figure 4.15: Simulated Entry Trajectory for First Scenario year 2000 vx.

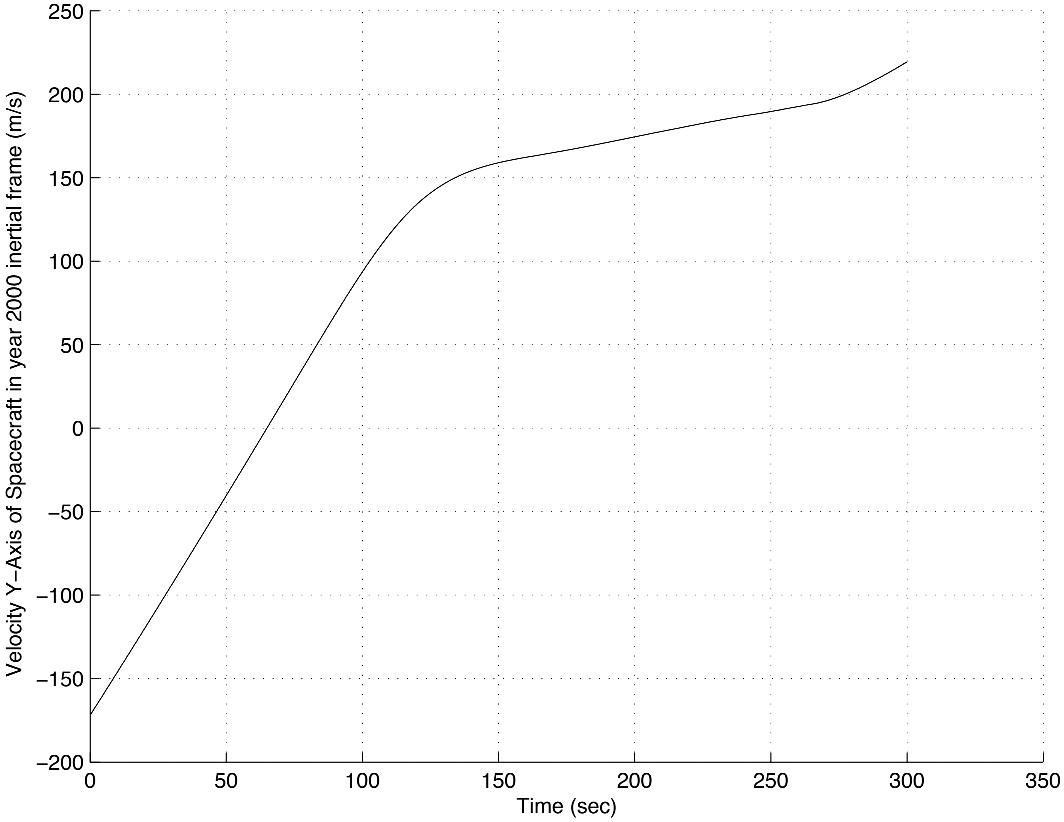

Figure 4.16: Simulated Entry Trajectory for First Scenario year 2000 vy.

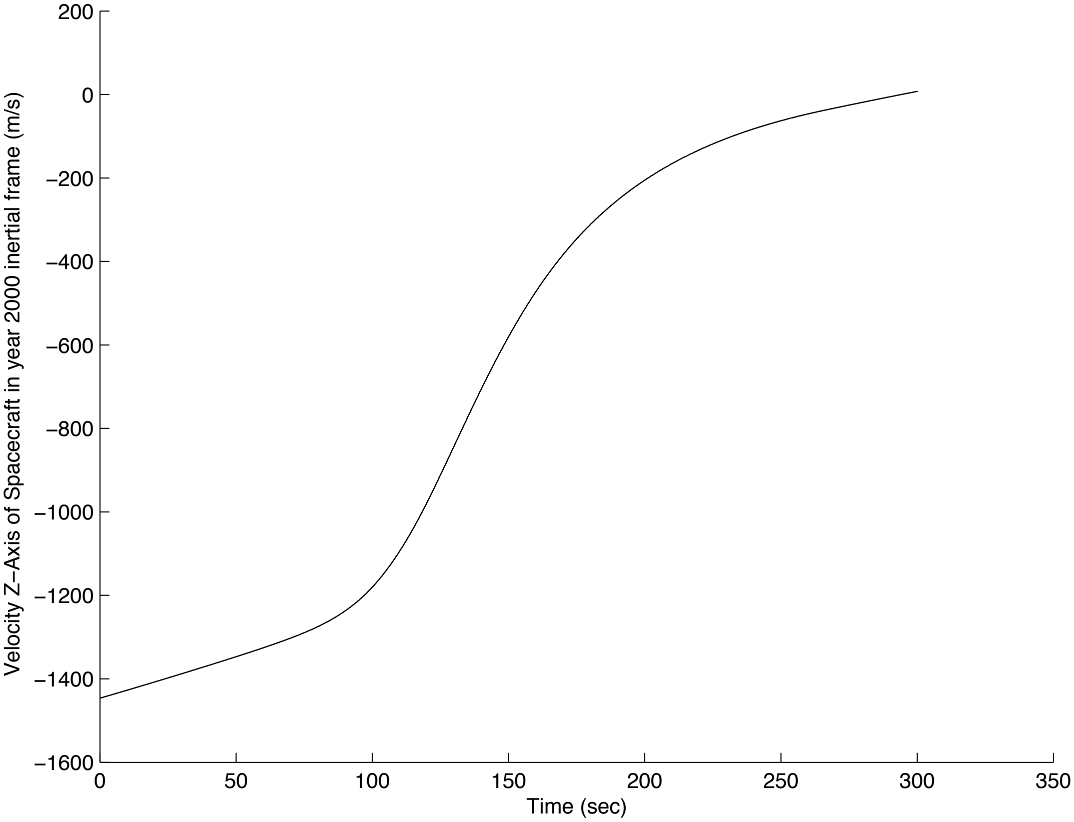

Figure 4.17: Simulated Entry Trajectory for First Scenario year 2000 vz.

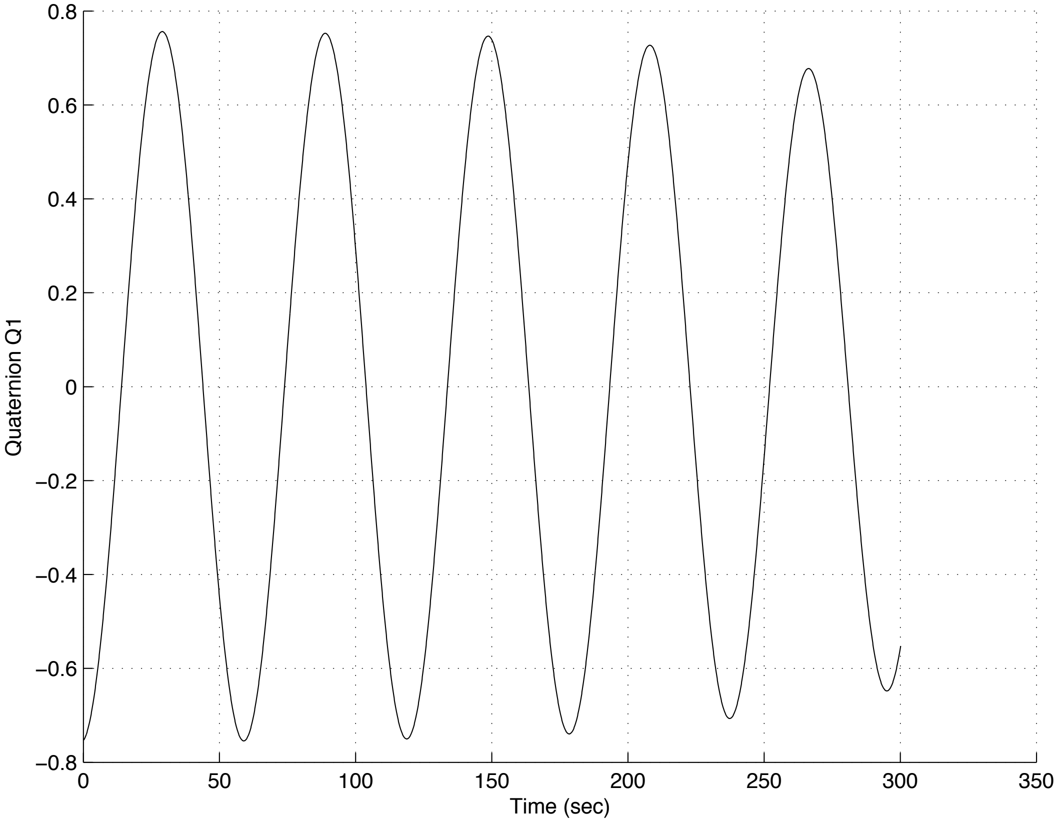

Figure 4.18: Simulated Entry Trajectory for First Scenario Quaternion Q1.

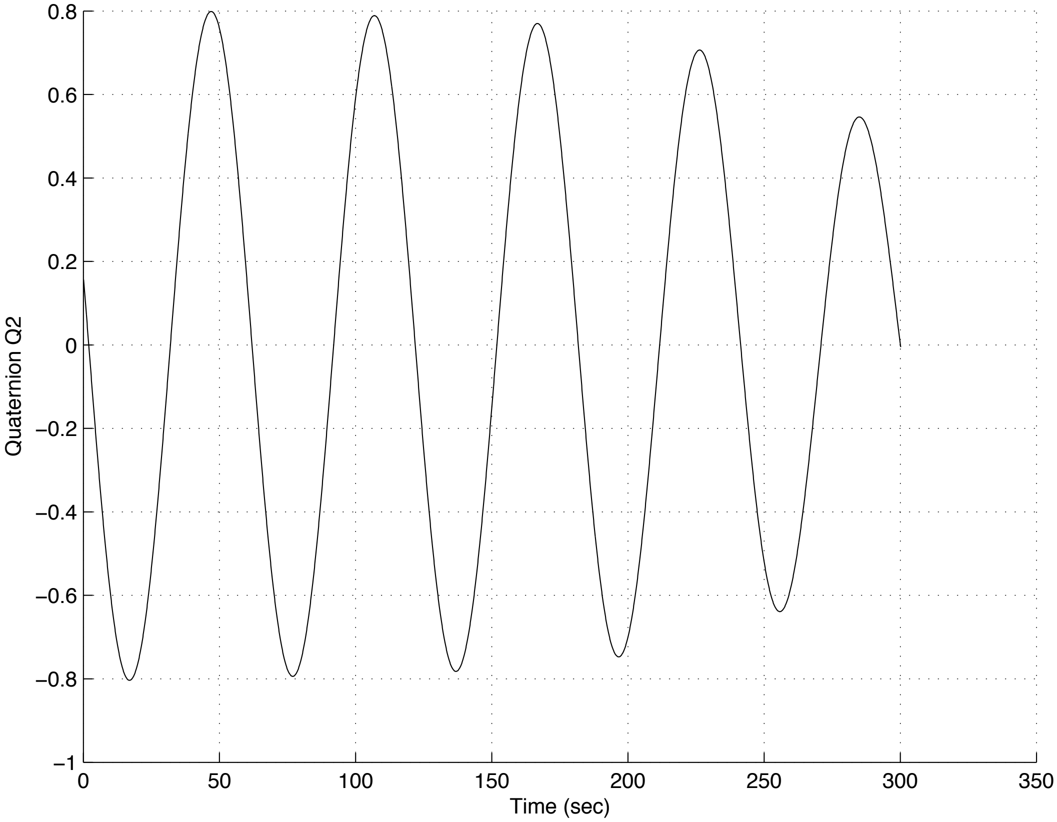

Figure 4.19: Simulated Entry Trajectory for First Scenario Quaternion Q2.

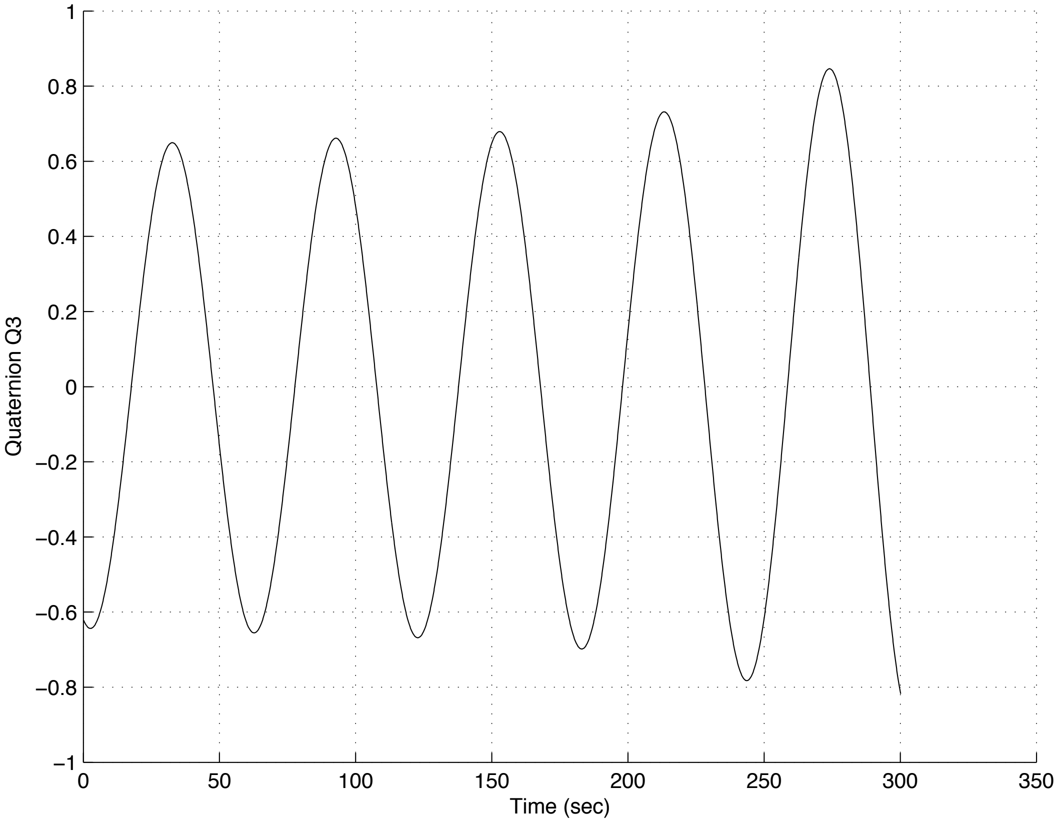

Figure 4.20: Simulated Entry Trajectory for First Scenario Quaternion Q3.

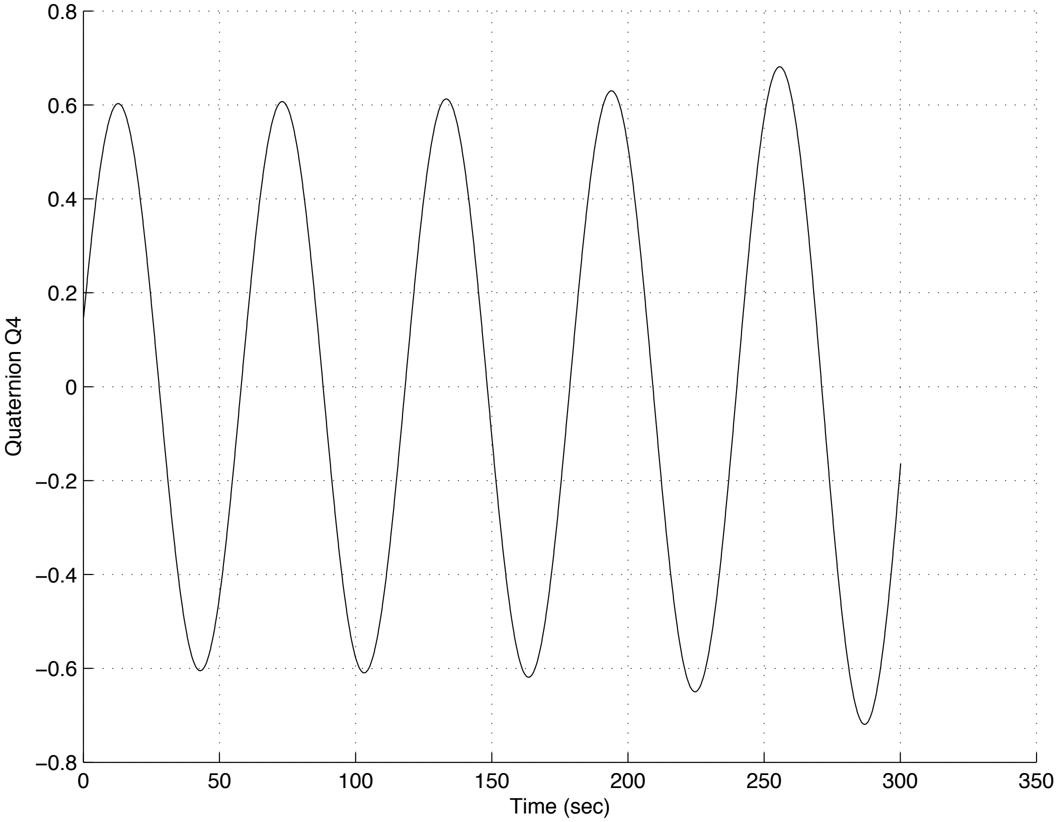

Figure 4.21: Simulated Entry Trajectory for First Scenario Quaternion Q4.

4.4.1.2 Second Scenario

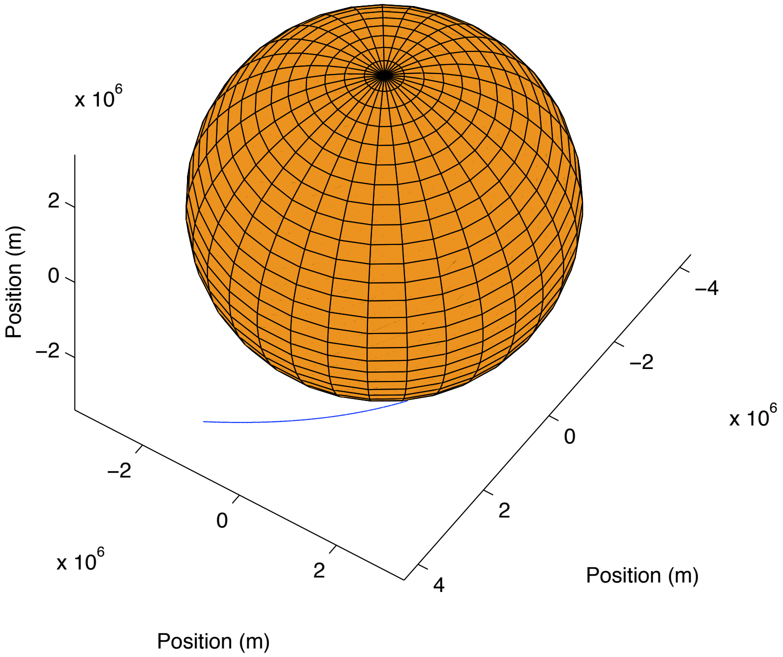

Figure 4.22: Entry Trajectory for Second Scenario.

Based on cruise stage separation data from the anticipated MSL entry the reference trajectory is created using the same dynamics as were found applicable for the MER entry.

The simulation of the second scenario trajectory is shown in Fig. (4.22) through Fig. (4.28).

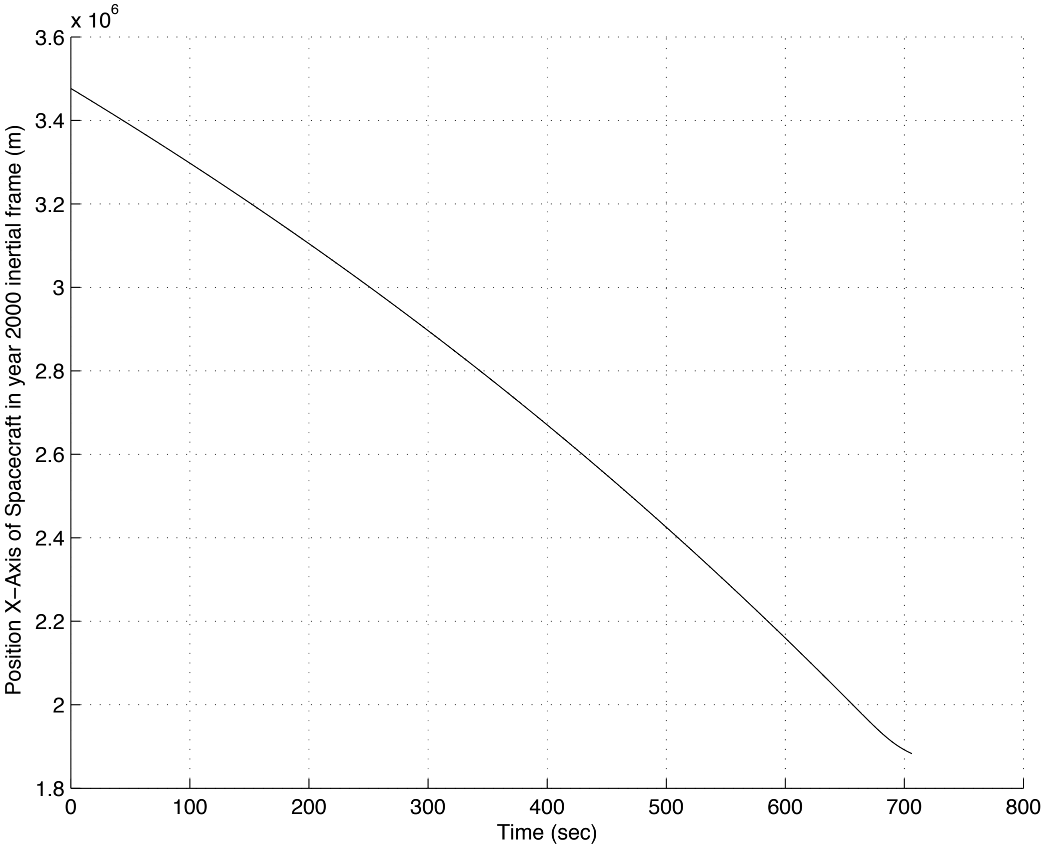

Figure 4.23: Simulated Entry Trajectory for Second Scenario year 2000 x.

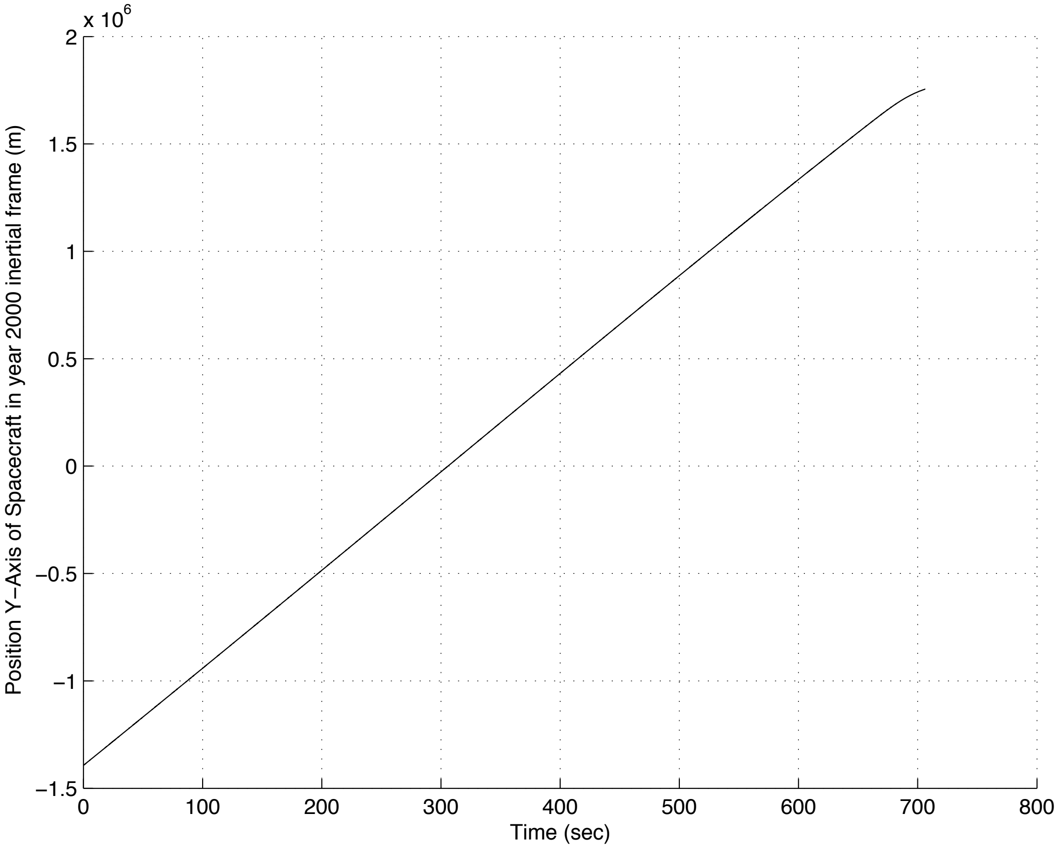

Figure 4.24: Simulated Entry Trajectory for Second Scenario year 2000 y.

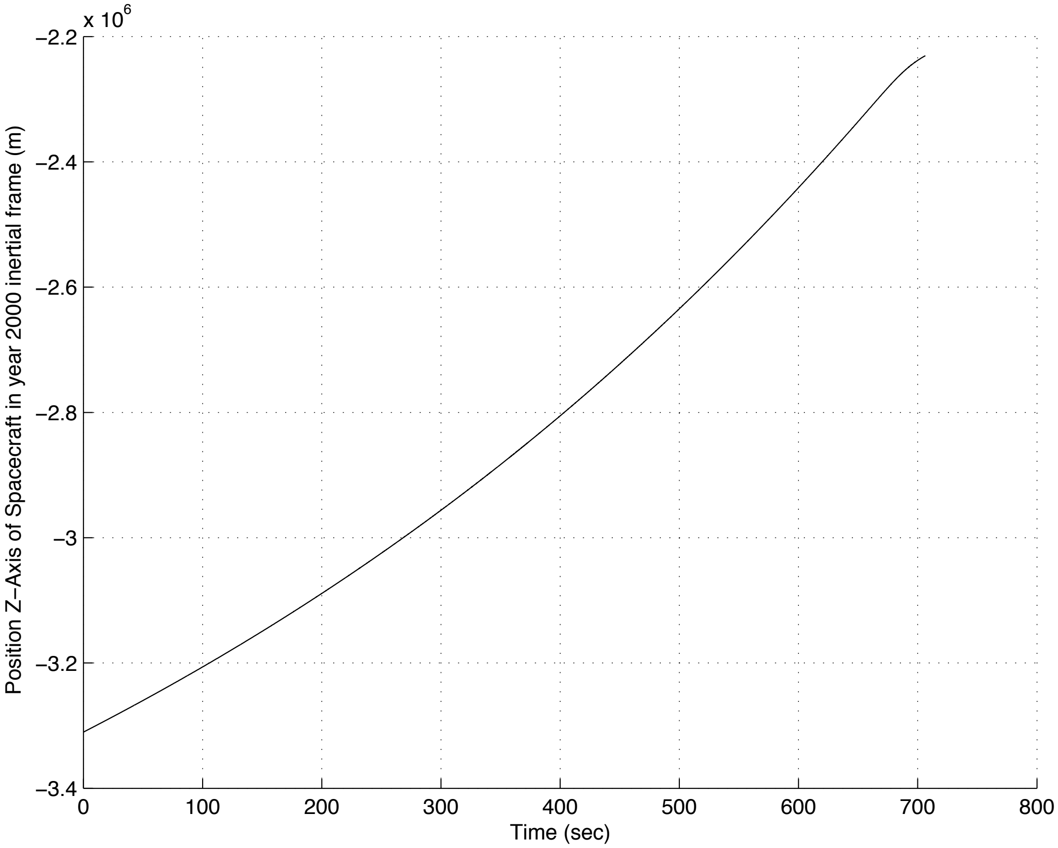

Figure 4.25: Simulated Entry Trajectory for Second Scenario year 2000 z.

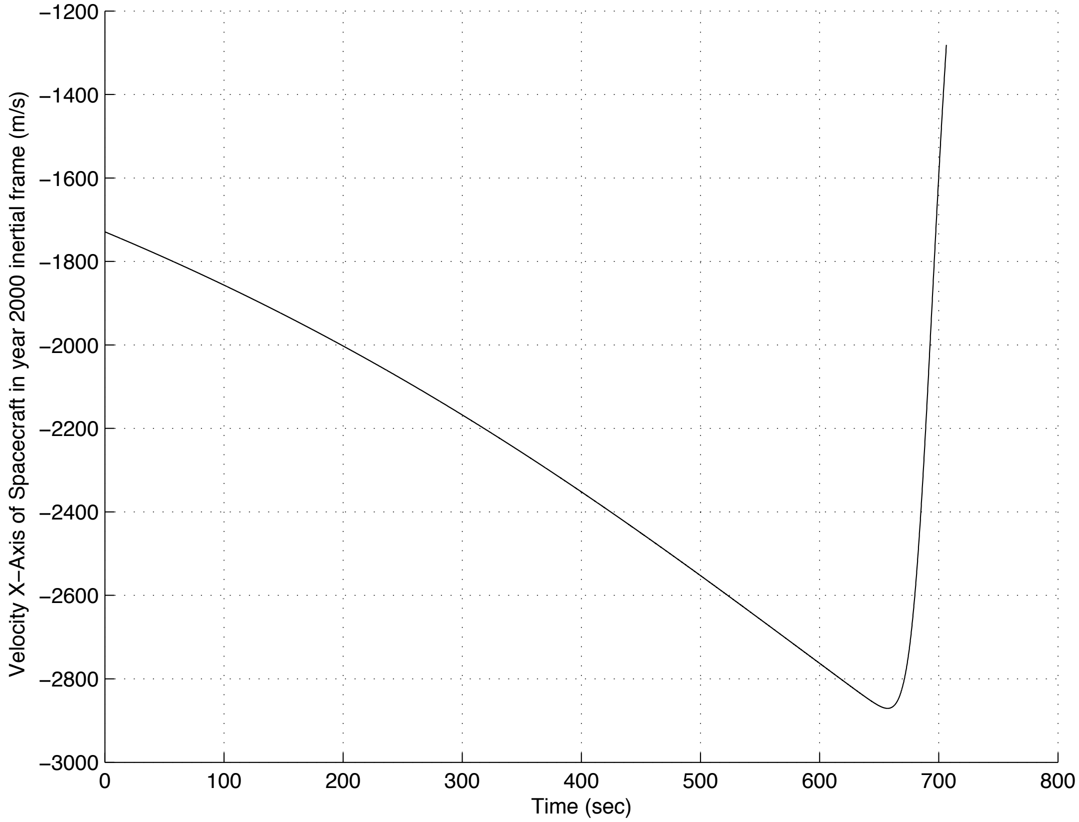

Figure 4.26: Simulated Entry Trajectory for Second Scenario year 2000 vx.

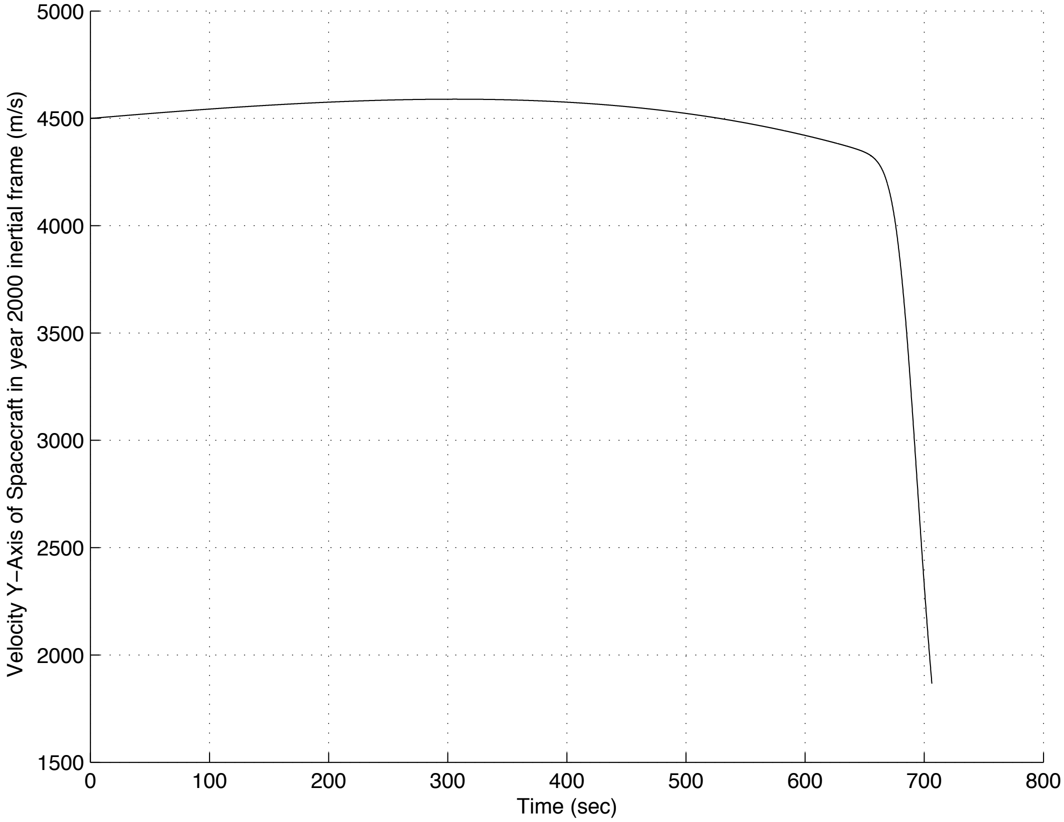

Figure 4.27: Simulated Entry Trajectory for Second Scenario year 2000 vy.

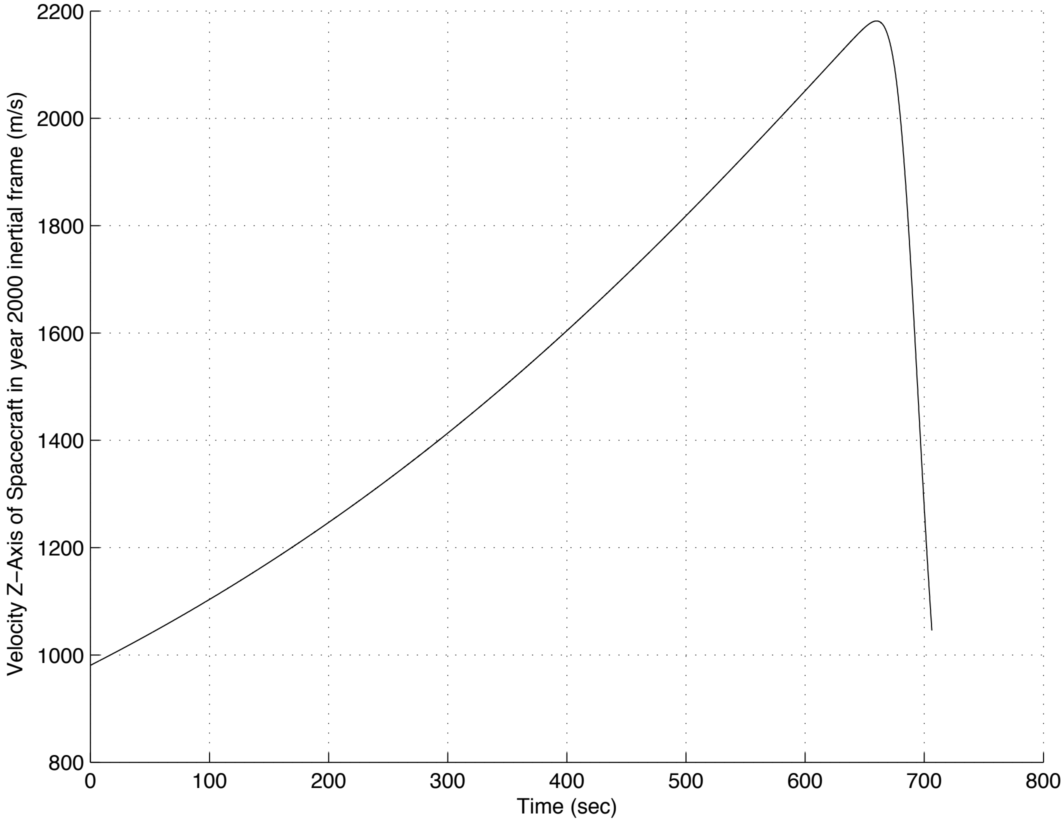

Figure 4.28: Simulated Entry Trajectory for Second Scenario year 2000 vz.

4.4.2 MER Trajectory Reconstruction

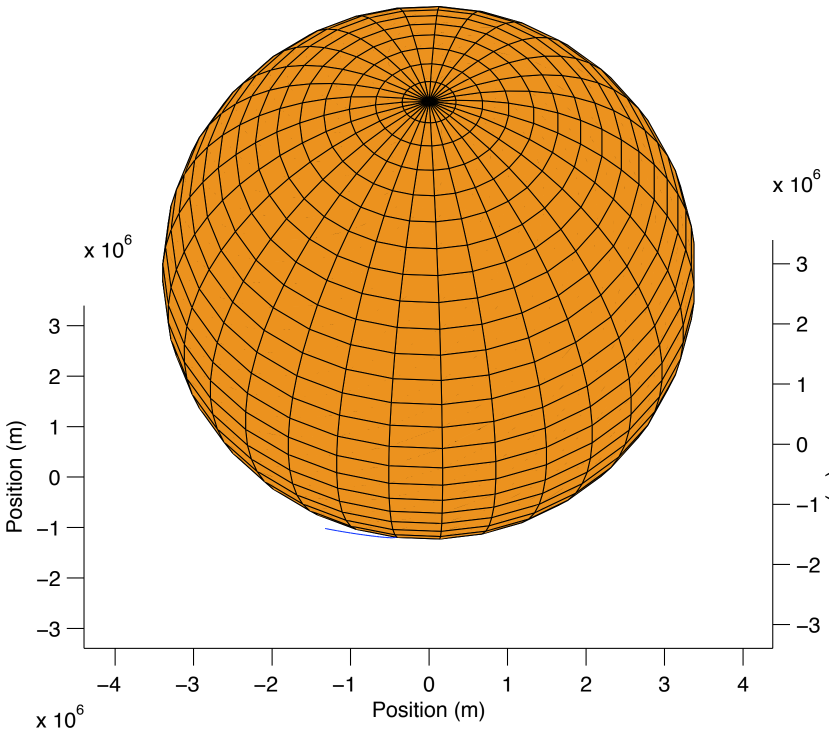

Figure 4.29: MER-A Data Dead Reckoning.

With the MER Trajectory data and IMU measurements available the trajectory can be reconstructed.

At each time step during the simulation the IMU data is taken and based on the knowledge of the state and the measurement model the new spacecraft state can be calculated. This process of dead reckoning the IMU data is commonly used for state determination of vehicles with IMUs.

The resulting MER-A trajectory as determined through dead reckoning the IMU data is shown in Fig. (4.29) through Fig. (4.35). The reconstruction matches the simulated trajectory.

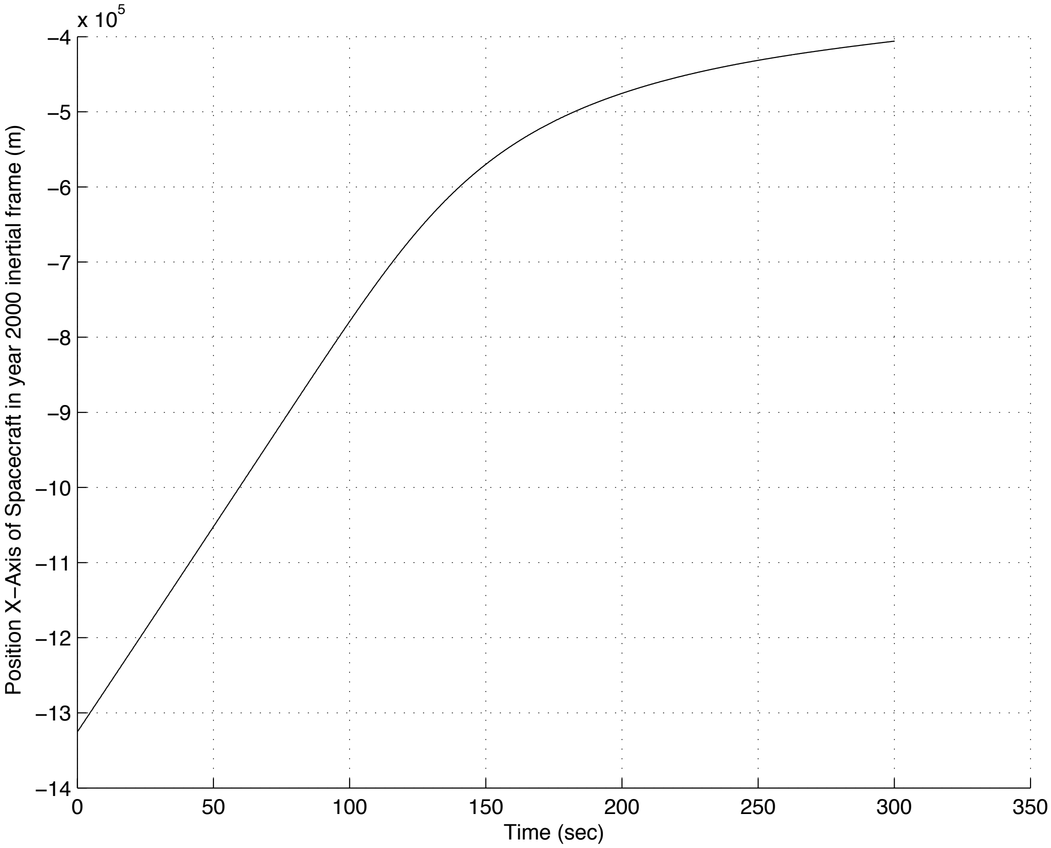

Figure 4.30: MER-A Data Dead Reckoning year 2000 x position.

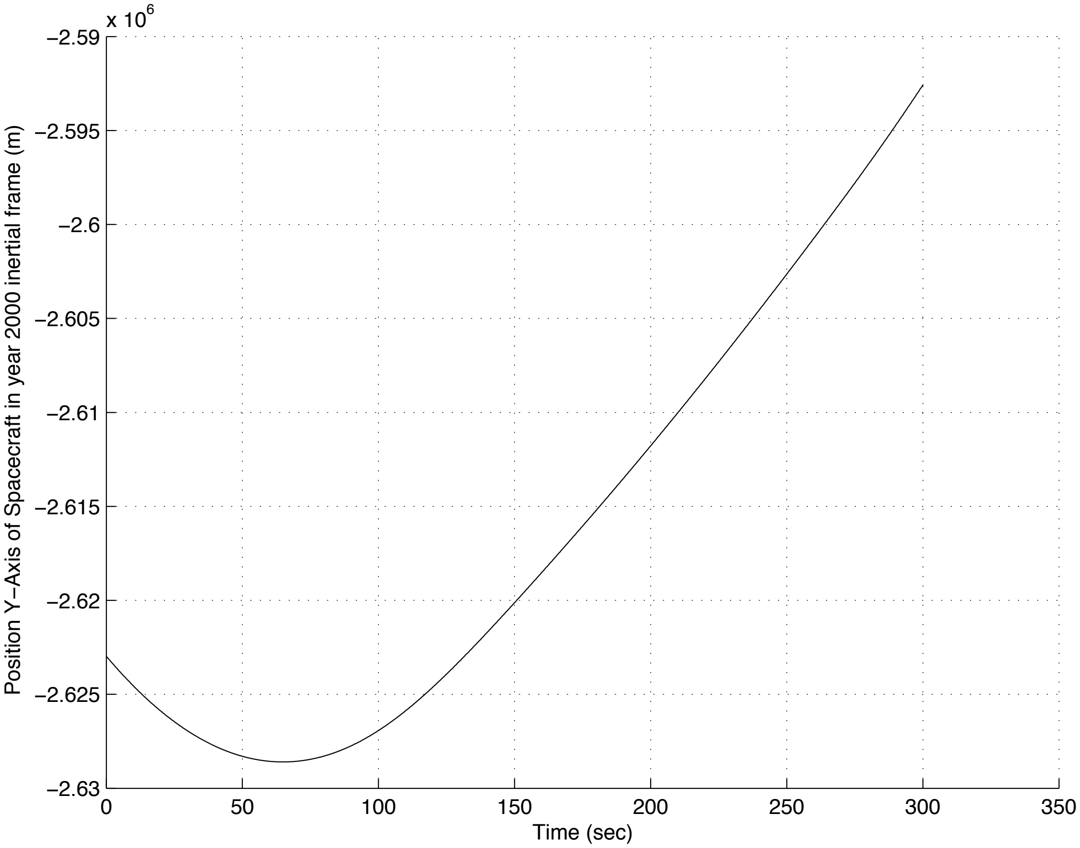

Figure 4.31: MER-A Data Dead Reckoning year 2000 y position.

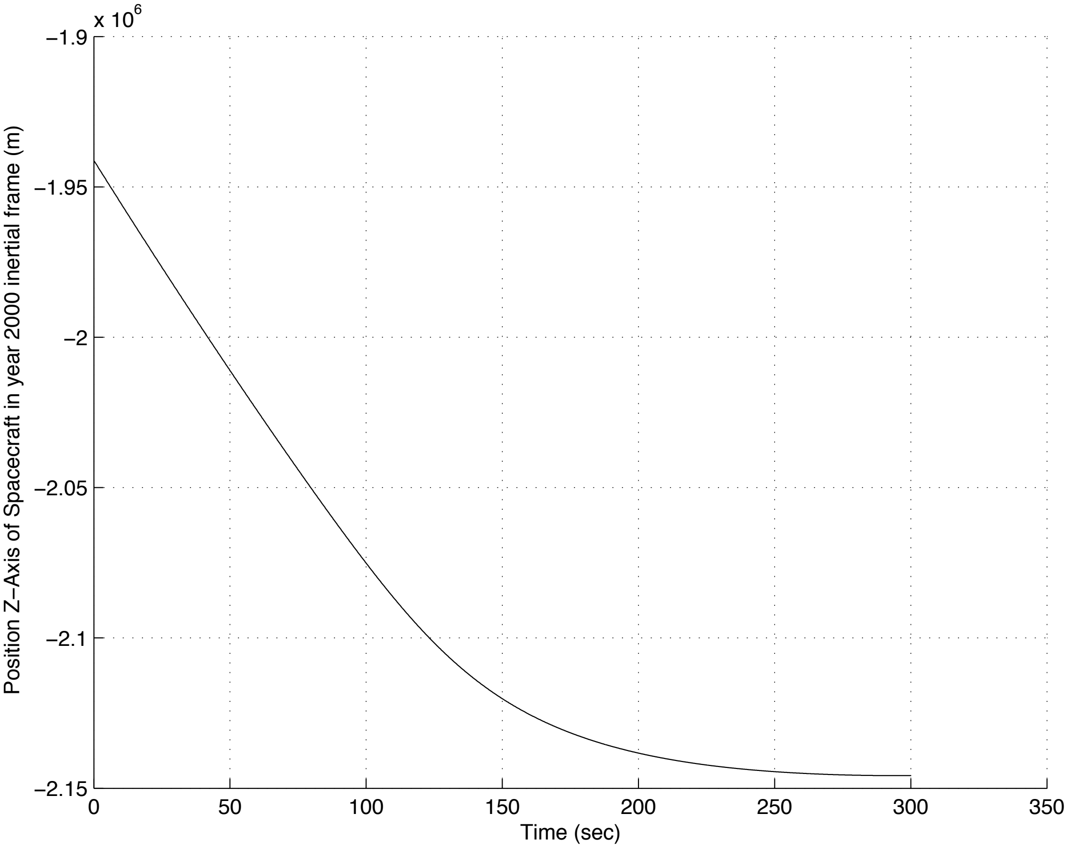

Figure 4.32: MER-A Data Dead Reckoning year 2000 z position.

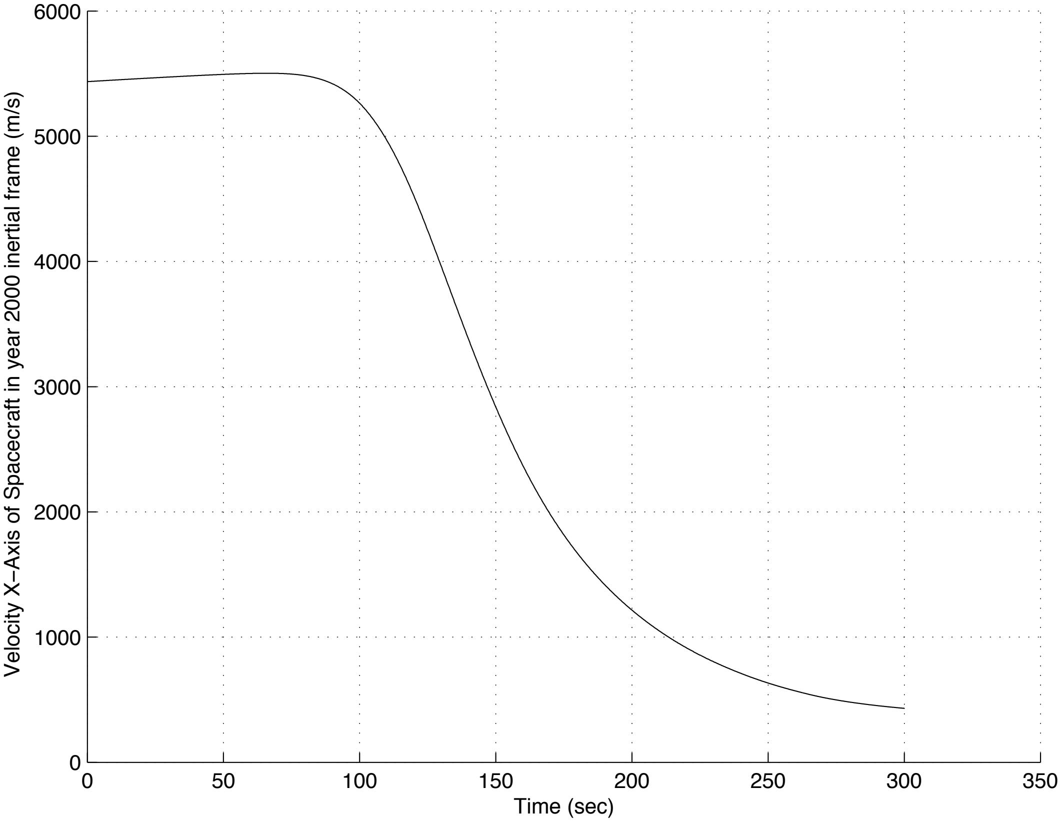

Figure 4.33: MER-A Data Dead Reckoning year 2000 x velocity.

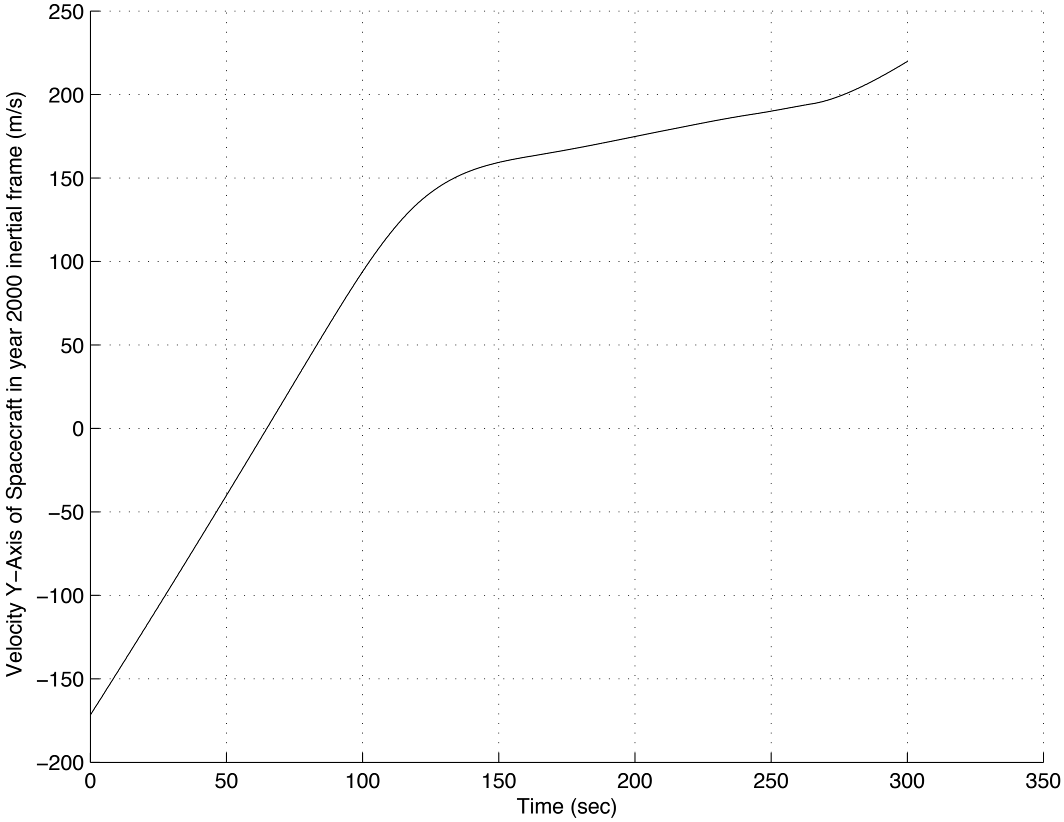

Figure 4.34: MER-A Data Dead Reckoning year 2000 y velocity.

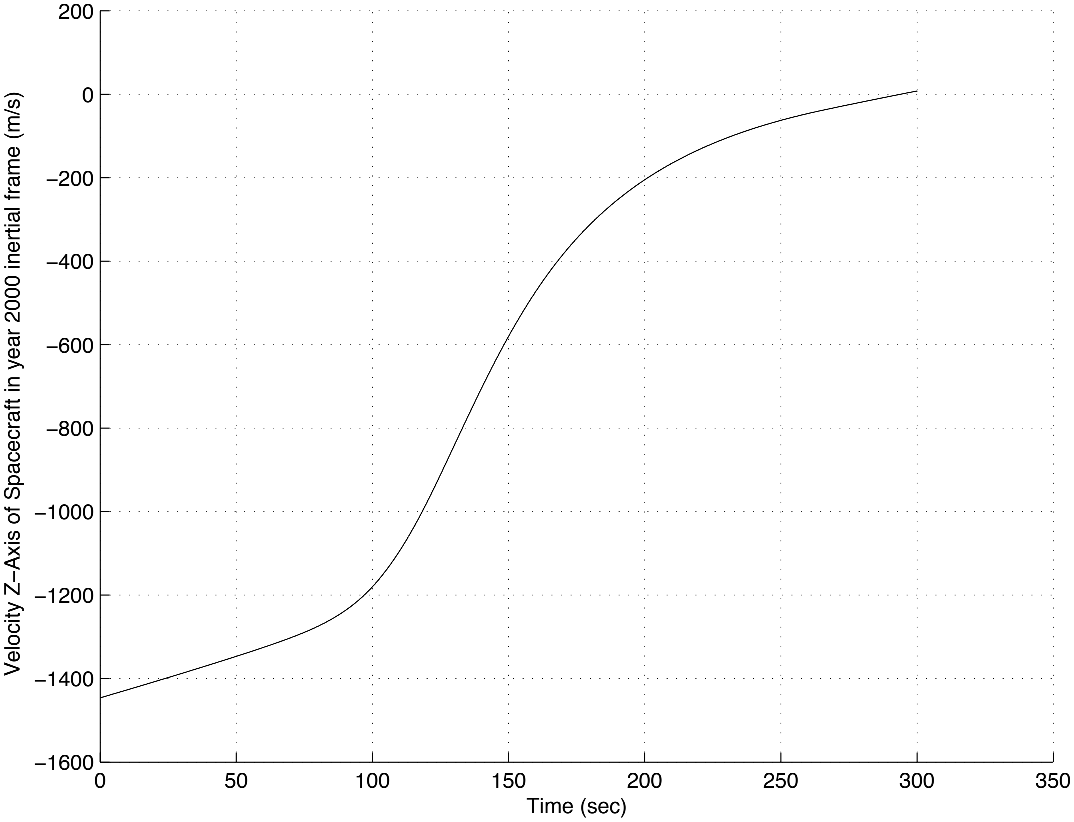

Figure 4.35: MER-A Data Dead Reckoning year 2000 z velocity.

Chapter 5

Simulation Results

It is of great interest to know how robust the proposed navigation concept is to poor knowledge of the spacecraft state and the changing external environment. The first step in the design process is to apply appropriate filter tuning to a single filter operating in a nominal atmospheric environment. To accomplish this, the process noise is modified and a *Monte Carlo* analysis is performed for the chosen settings to evaluate the performance of the single filter during EDL.

It is known that the density variations at Mars are so great that a single filter cannot provide excellent state estimates over the wide range of expected density variations. In order to have a more robust system, a filter bank is created which has several filters running simultaneously. Implementation of filters covers a wide range of environment parameters and is expected to increase the navigation robustness to deviations in the predicted environment compared to the use of a single filter.

In order to analyze the filter performance a *Monte Carlo* analysis is performed. This method gives an understanding of the ability of the architecture to operate successfully from an initial state represented as a probability density.

5.1 Mars Entry Navigation Analysis

5.1.1 Filter Performance

The sigma point Kalman filter (SPKF) performance for the task of navigation during the hypersonic Mars entry is presented here. As the only available observation comes from the IMU during this segment of the entry, a direct comparison against the purely dead-reckoned trajectory reveals how well the filter is operating.

The plots presented in this chapter only show the velocity portion of the state which we expect to be directly affected by the IMU measurement. In the plots, the blue line represents the estimation error of the filter, the magenta line represents the filter aposteiori covariance and the red line is the dead-reckoned velocity error.

5.1.1.1 Low Noise Environment

Using the filter with the measurement history with low noise gives an estimate of the spacecraft velocity as shown in Fig. (5.1) through Fig. (5.3). For one particular dispersed filter run the velocity estimate is shown. A dispersed run is used here instead of the average *Monte Carlo* run in order to better compare it to the dead-reckoning of the same run. The dead-reckoned result is also presented in the plots.

It can be seen that with low measurement noise, the dead-reckoned trajectory has a smaller deviation from the actual state than the filter estimate beginning with the point where the aerodynamic forces become significant. During the time of high dynamics, the filter dead-reckoned state is more accurate. This is caused by the uncertainties in the filter having an effect on the filter estimate. The dead-reckoning performs better in the situation shown due to the fact that the initial state errors were very small presumably because of extensive DSN tracking. Also the particular case of low noise represents the use of a highly accurate and expensive IMU. With no errors in the initial state and an error-free IMU, the dead-reckoning process is a successful navigation tool and the filter-based navigation deviates further from the truth due to uncertainties in the model parameters. As seen in Sec 5.1.1.2, this situation does not extend to the medium and high noise environments.

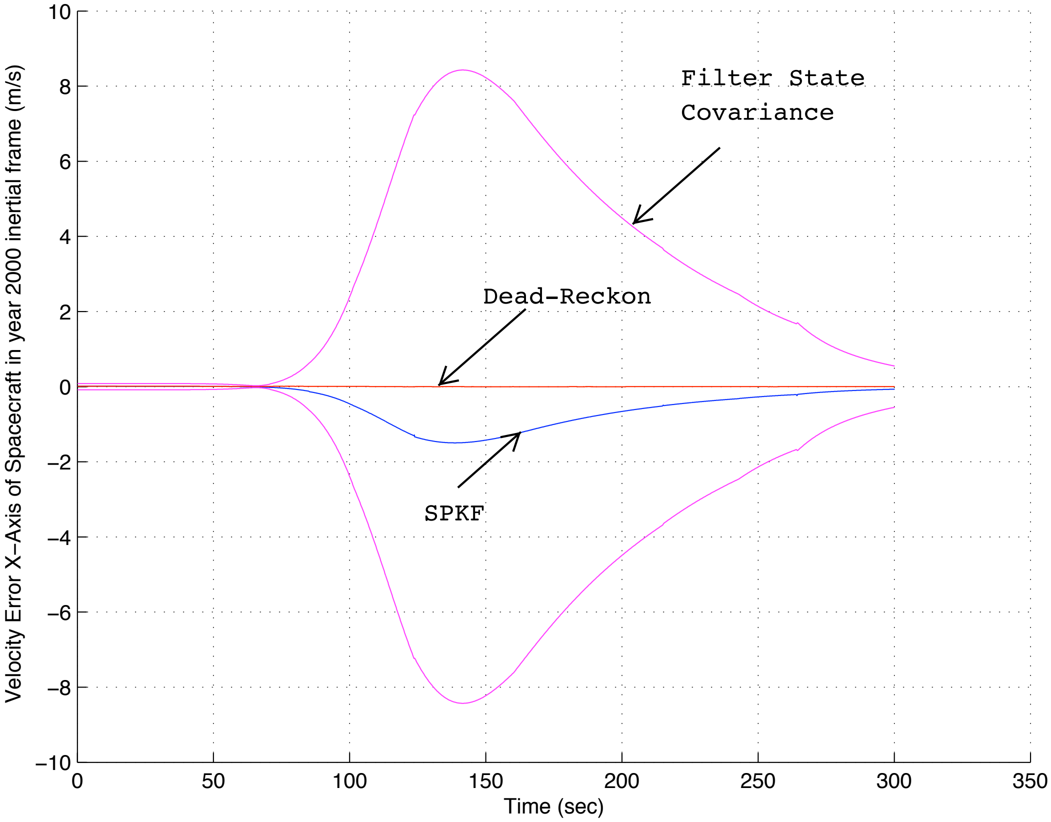

Figure 5.1: SPKF Velocity Error for Low Measurement Noise year 2000 x.

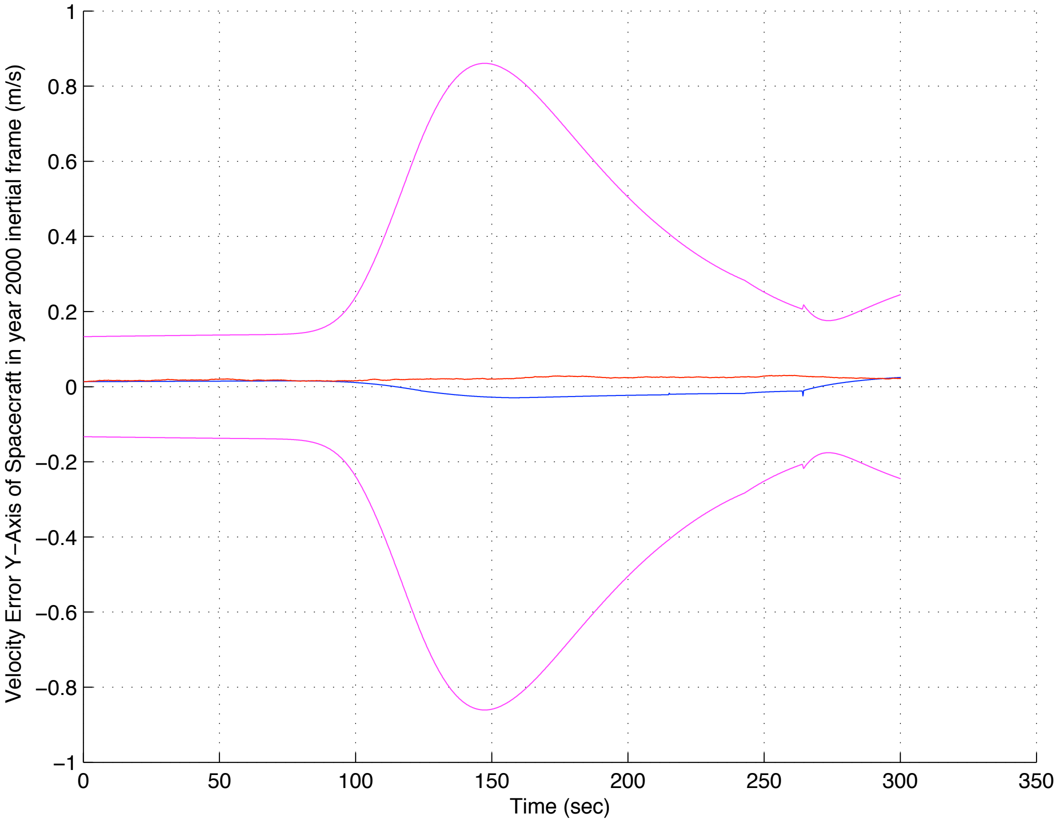

Figure 5.2: SPKF Velocity Error for Low Measurement Noise year 2000 y.

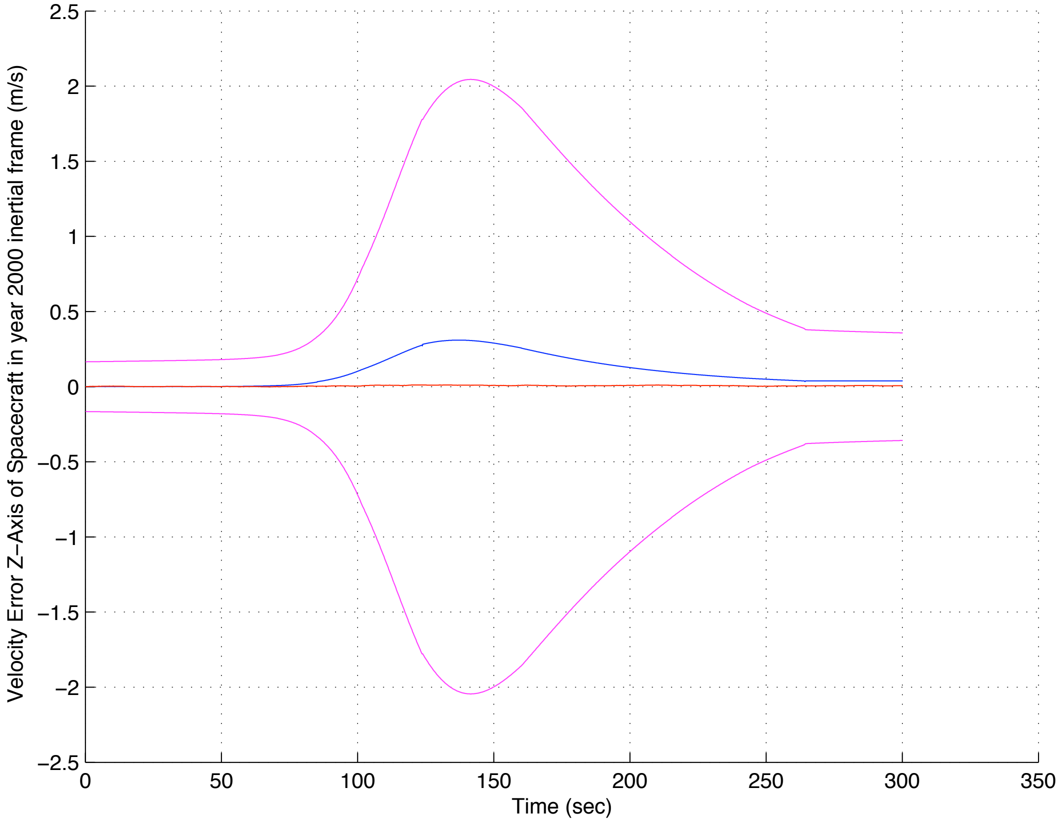

Figure 5.3: SPKF Velocity Error for Low Measurement Noise year 2000 z.

5.1.1.2 Medium Noise Environment

Using the filter with the measurement history with medium noise gives an estimate of the spacecraft velocity as shown in Fig. (5.4) through Fig. (5.6).

From the beginning the dead-reckoning estimate is poor. The navigation filter performs better than the dead-reckoning. It can be seen that with medium noise the dead-reckoned trajectory deviates away from the actual state farther than the filter estimate. While at the time of high dynamic activity, the dead-reckoned trajectory in the X-axis temporarily is closer to the truth, the filter recovers from that and gives a more accurate estimate when approaching filter parachute. In both the Y- and Z-axis, the filter clearly performs better than the dead-reckoning solution. The reason for the the temporary deviation from the actual state in the filter is the uncertainty of the state during high dynamic activity. This can be seen in the covariance growing quickly after the atmosphere is first encountered and reaching a maximum at the time when the highest forces act on the spacecraft as seen in the measurement history in Fig. (4.6).

In the medium noise environment, the advantage of using a model-based navigation concept becomes evident. The dead-reckoning of the IMU data in this case is less reliable than the SPKF estimate. In addition to the better estimation of the spacecraft state, the filter also provides a covariance estimate giving information on the certainty of the information. Also, the noise clearly effects the dead-reckoning estimate while the filter is able to mitigate the noise and appropriately expands the covariance information during the time of high dynamic activity. In all three axes the filter follows the truth well.

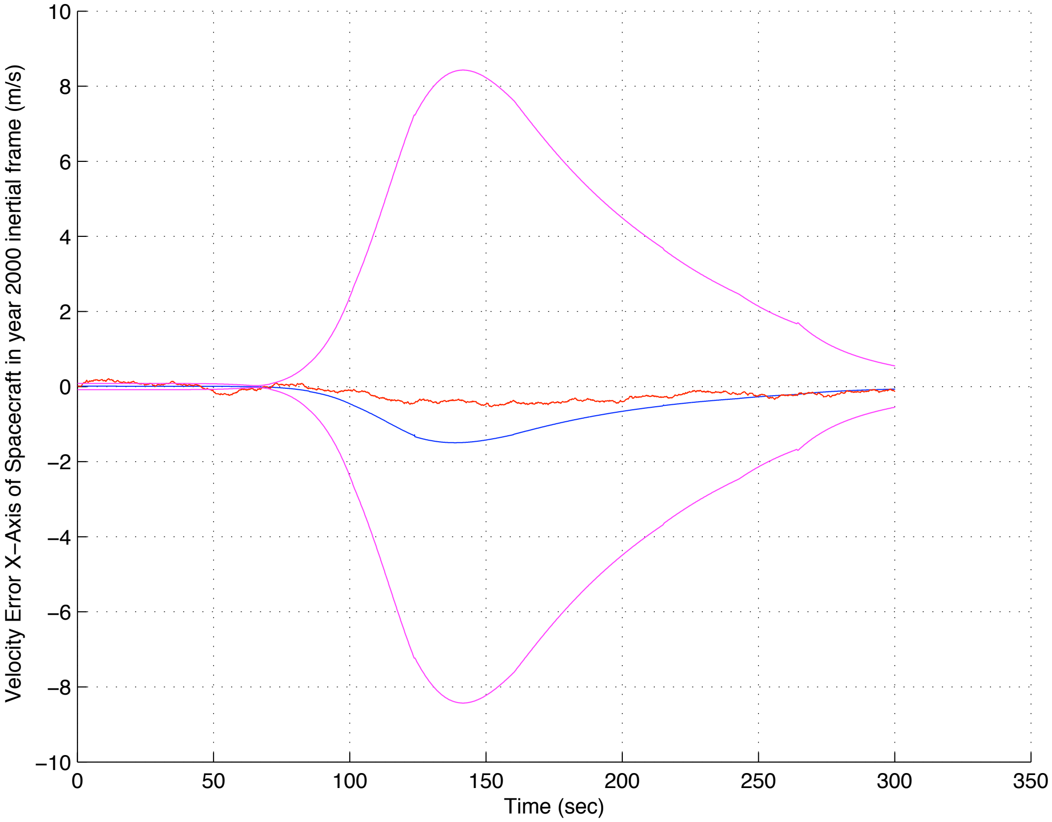

Figure 5.4: SPKF Velocity Error for Medium Measurement Noise year 2000 x.

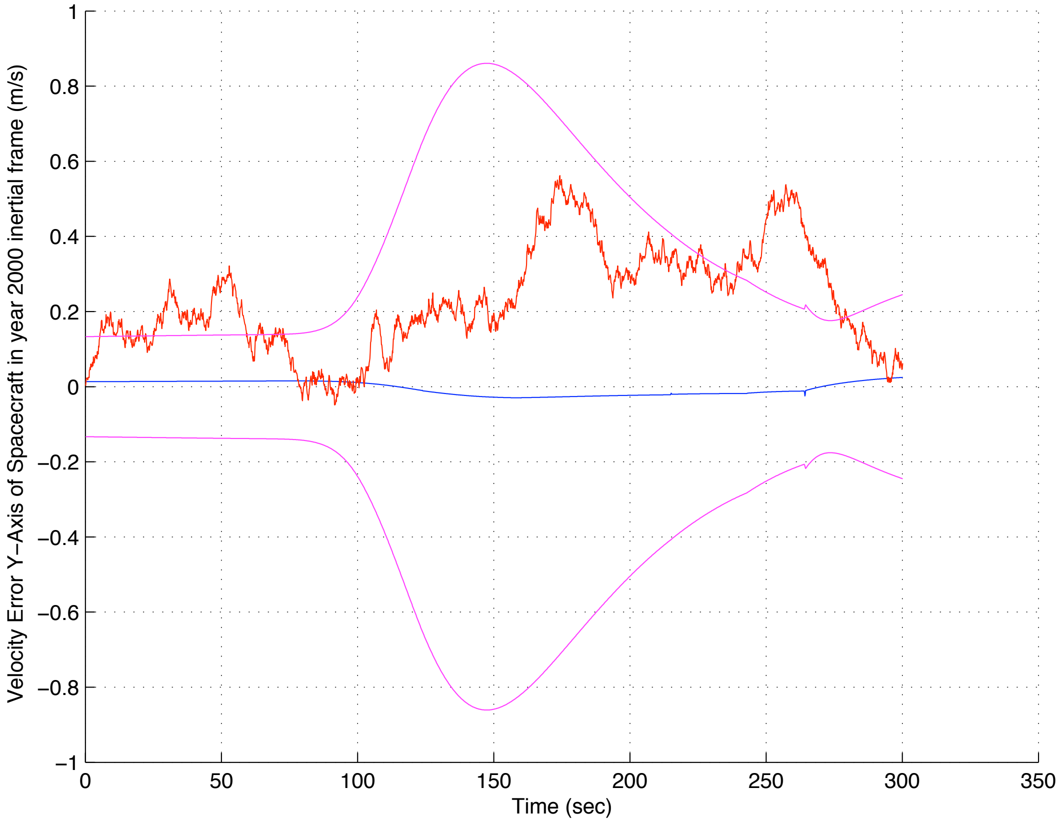

Figure 5.5: SPKF Velocity Error for Medium Measurement Noise year 2000 y.

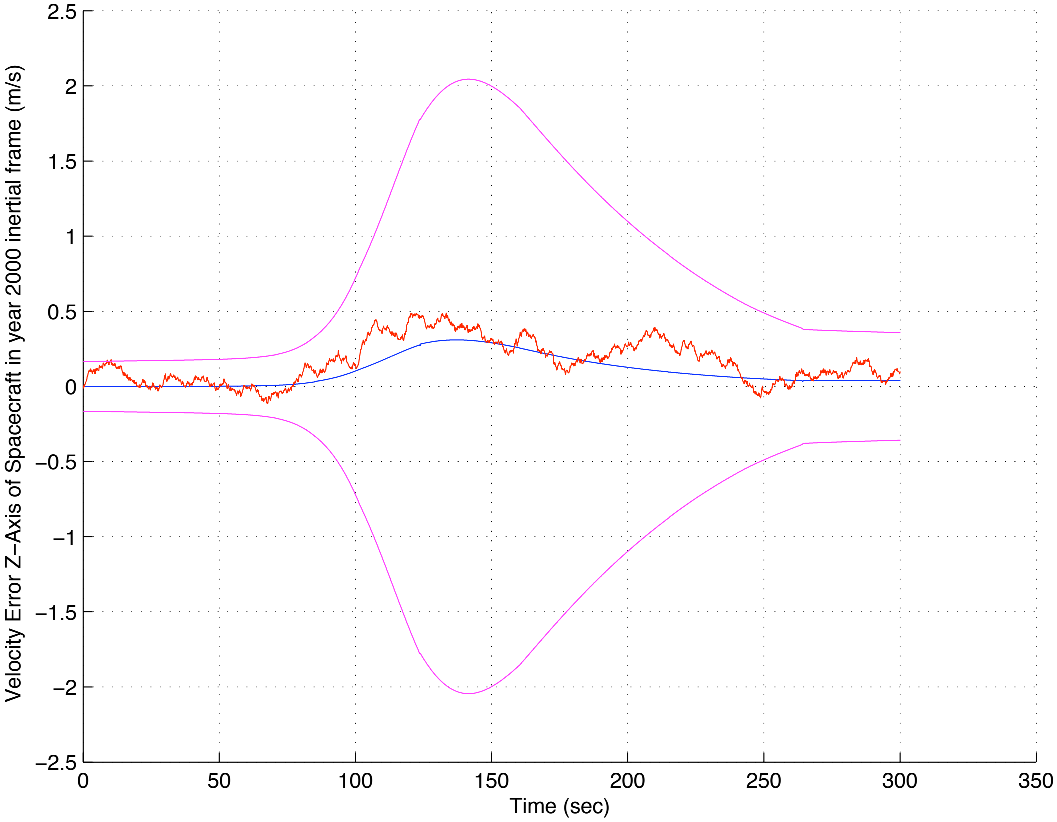

Figure 5.6: SPKF Velocity Error for Medium Measurement Noise year 2000 z.

5.1.1.3 High Noise Environment

Using the filter with the measurement history with high noise gives an estimate of the spacecraft velocity as shown in Fig. (5.7) through Fig. (5.9).

In the presence of high measurement noise, the filter based estimate significantly outperforms the dead reckoned trajectory. In all axes, the dead-reckoning is largely outside the covariance for the certainty of the filter estimate. The quality of the filter estimate is up to 10 times superior to the dead-reckoning of the IMU seen in Fig. (5.8). The model-based filter concept in this case is reliably giving an estimate that is largely unaffected by the noise. In fact, when comparing the filter estimates for the different noise levels, the estimate is essentially the same for low, medium and high noise. The filter is robust to the noise level and continues to give the same good state estimates in all environments. The dead-reckoning on the other hand continuously degrades in the presence of noise.

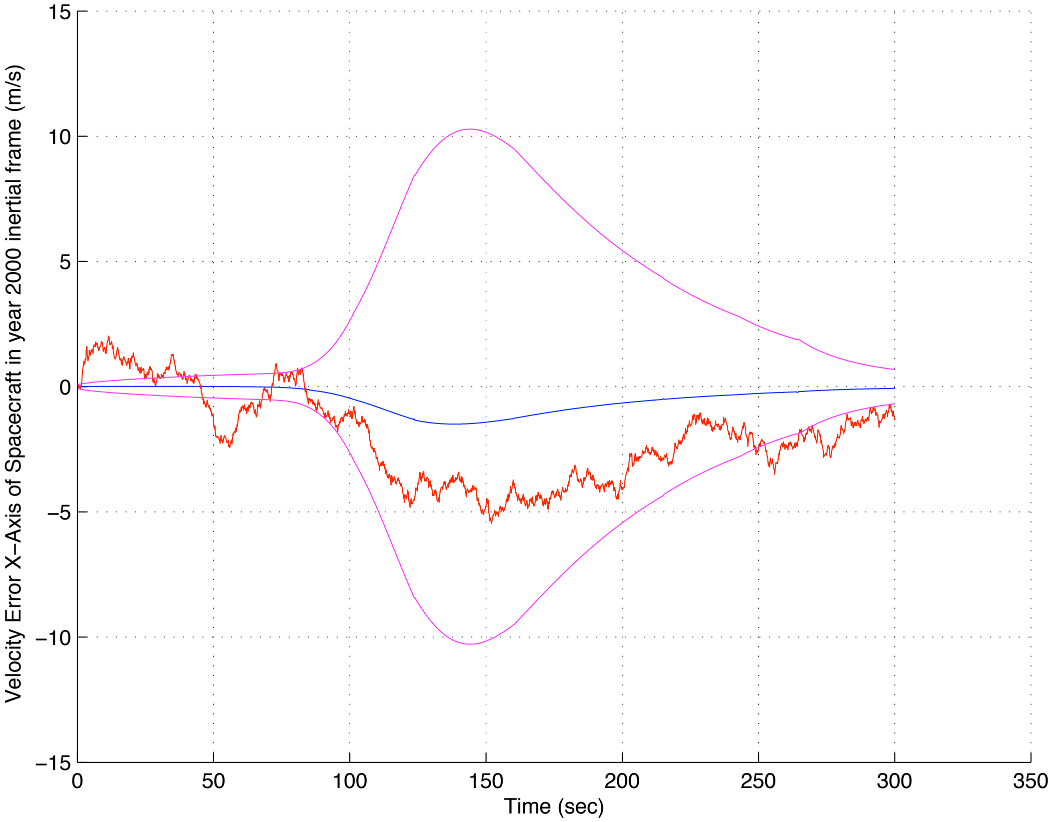

Figure 5.7: SPKF Velocity Error for High Measurement Noise year 2000 x.

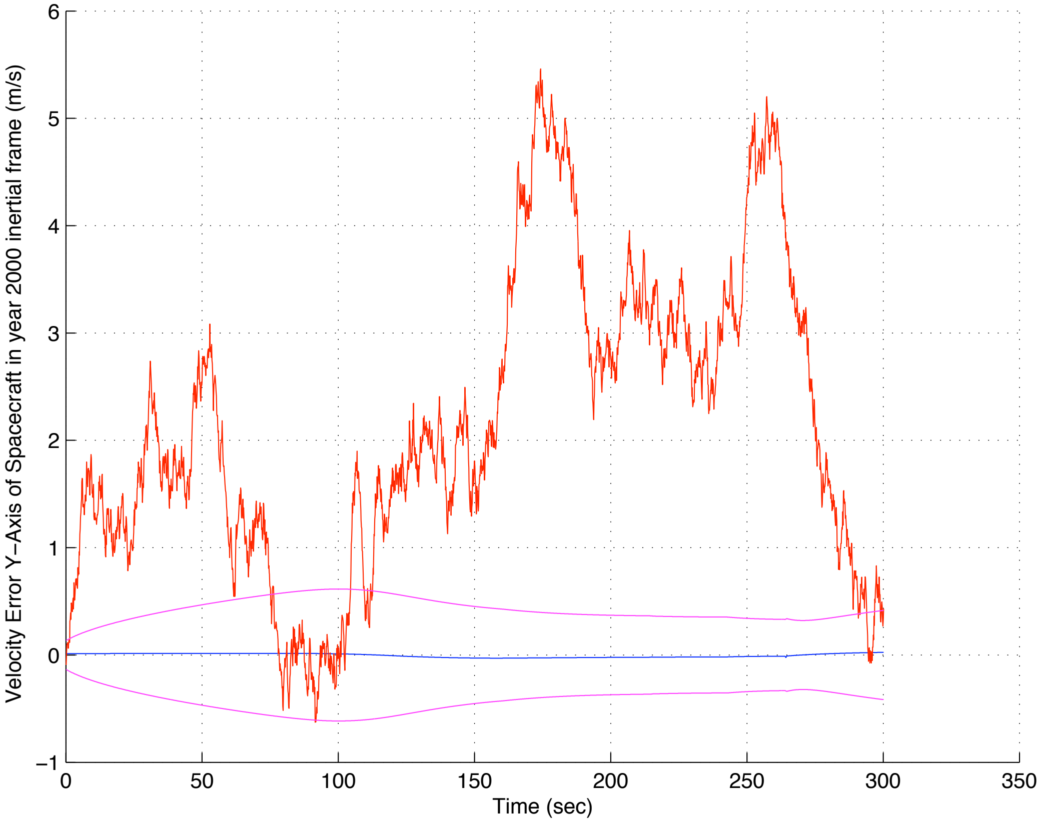

Figure 5.8: SPKF Velocity Error for High Measurement Noise year 2000 y.

Figure 5.9: SPKF Velocity Error for High Measurement Noise year 2000 z.

5.1.2 Monte Carlo Analysis

For 100 dispersed runs, the average results compared to the dead-reckoned solution illustrate the advantage of the proposed navigation architecture.

5.1.2.1 Low Noise Environment

In the presence of low noise the average Monte Carlo estimate is shown in Fig. (5.10) through Fig. (5.12).

For the average filter run the velocity estimate is shown. The result for the dead reckoned estimate and the filter estimate are essentially equivalent in this case.

As found for the individual run, the same trend is observed for the *Monte Carlo* analysis over a set of 100 simulated runs. In the low noise environment, a dead-reckoning estimate follows the actual state well. The filter also performs well and is able to reliably give a good estimate and a valid covariance describing the uncertainty of the state estimate. The filter estimate becomes slightly less accurate when atmospheric forces significantly affect the spacecraft. After the peak in the aerodynamics, the uncertainty declines and the filter estimate matches the actual trajectory again.

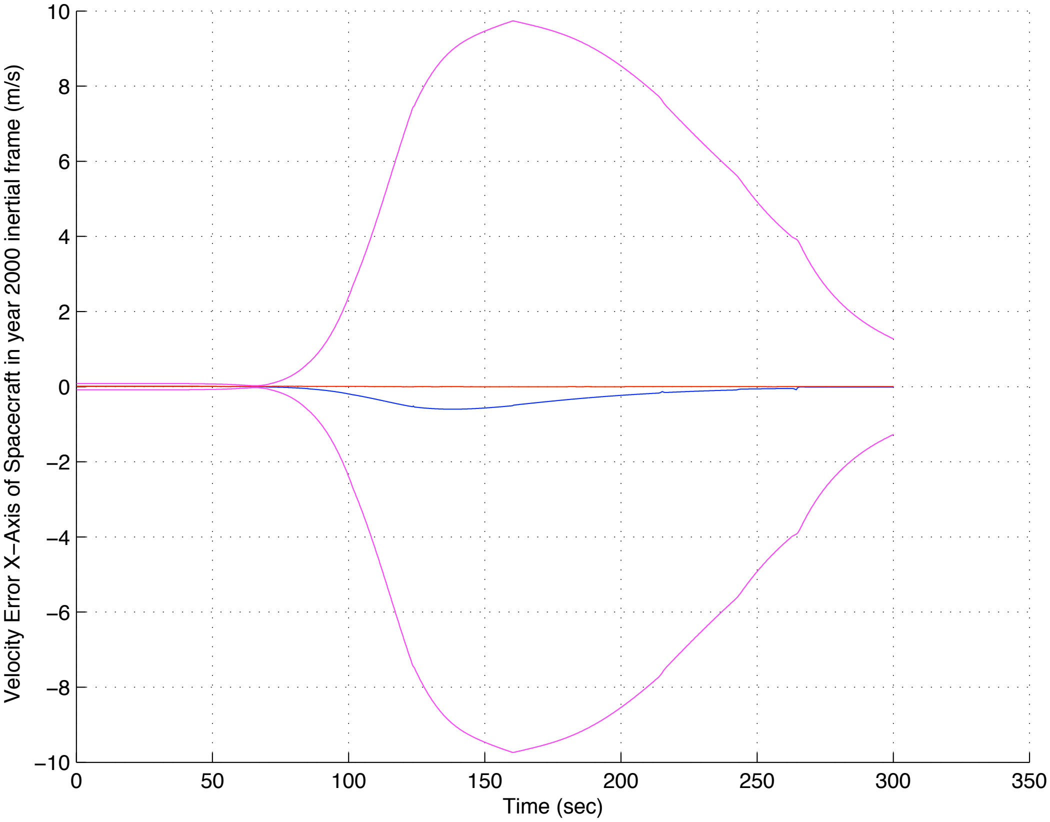

Figure 5.10: Monte Carlo average Velocity Error for Low Noise year 2000 x.

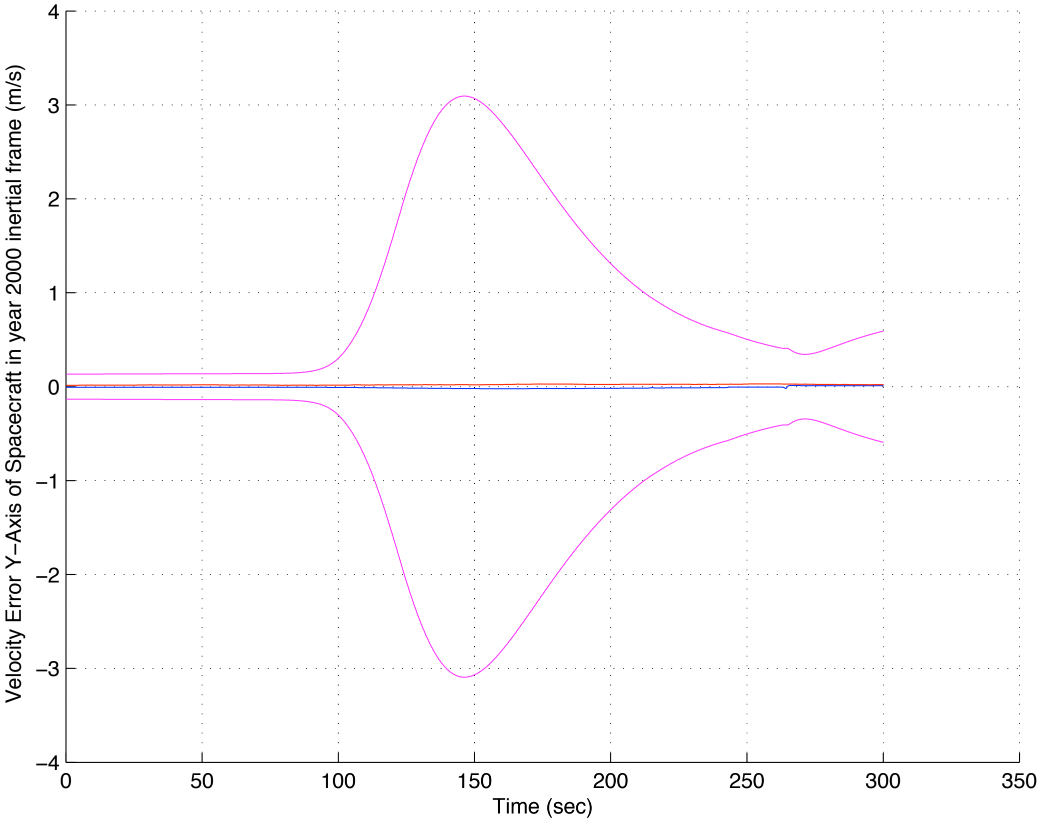

Figure 5.11: Monte Carlo average Velocity Error for Low Noise year 2000 y.

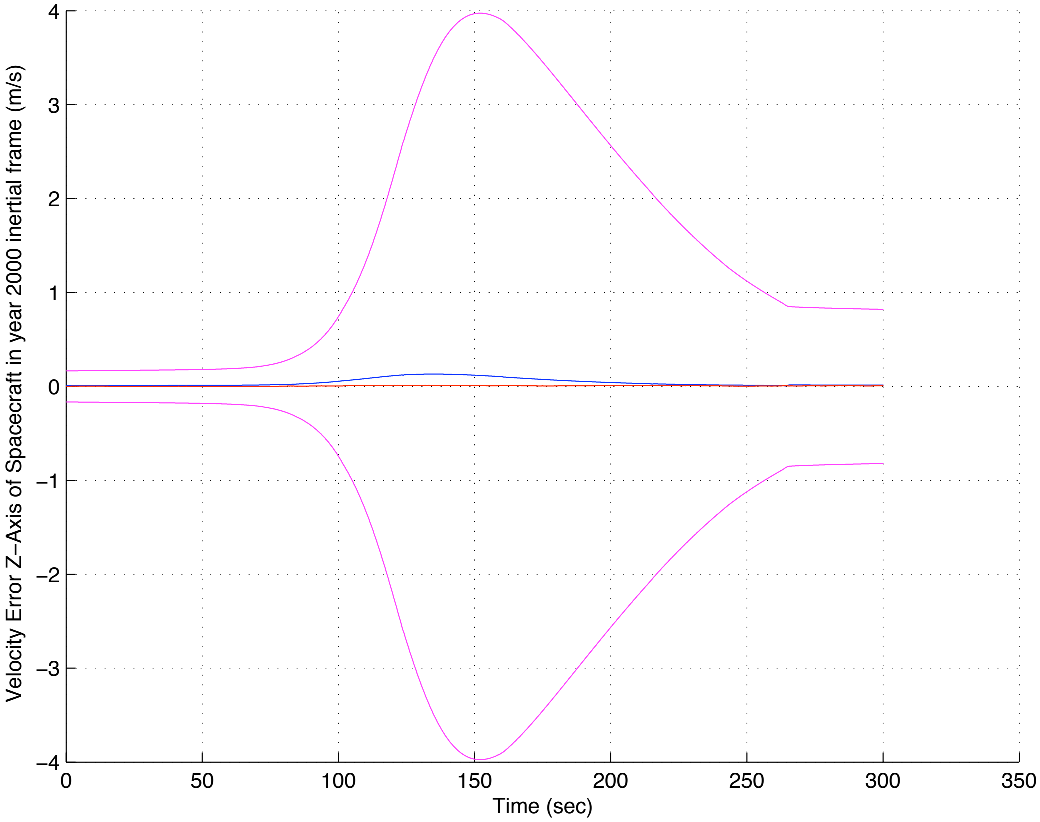

Figure 5.12: Monte Carlo average Velocity Error for Low Noise year 2000 z.

5.1.2.2 Medium Noise Environment

In the presence of medium noise the average Monte Carlo estimate is shown in Fig. (5.13) through Fig. (5.15).

The result for the filter estimate here is a clear improvement over the dead-reckoned trajectory. Considering the average *Monte Carlo* run it becomes clear that the filter estimate is superior to the dead-reckoning. The state is essentially estimated correctly throughout the hypersonic phase, again before the atmosphere begins to have an effect the estimate matches the truth. With the atmospheric dynamics introducing uncertainty the filter estimate is affected slightly, but recovers well in time and returns to matching the truth. At the same time the dead-reckoning process deviates from the truth and continues to provide no useful estimate.

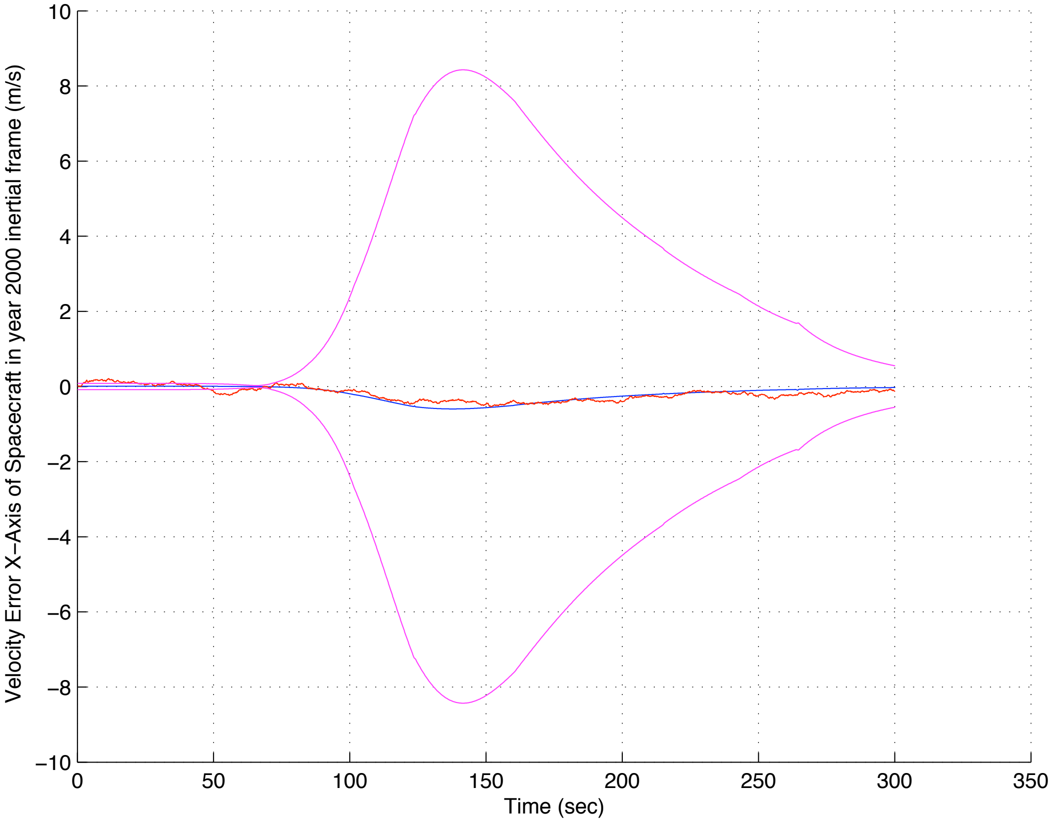

Figure 5.13: Monte Carlo average Velocity Error for Medium Noise year 2000 x.

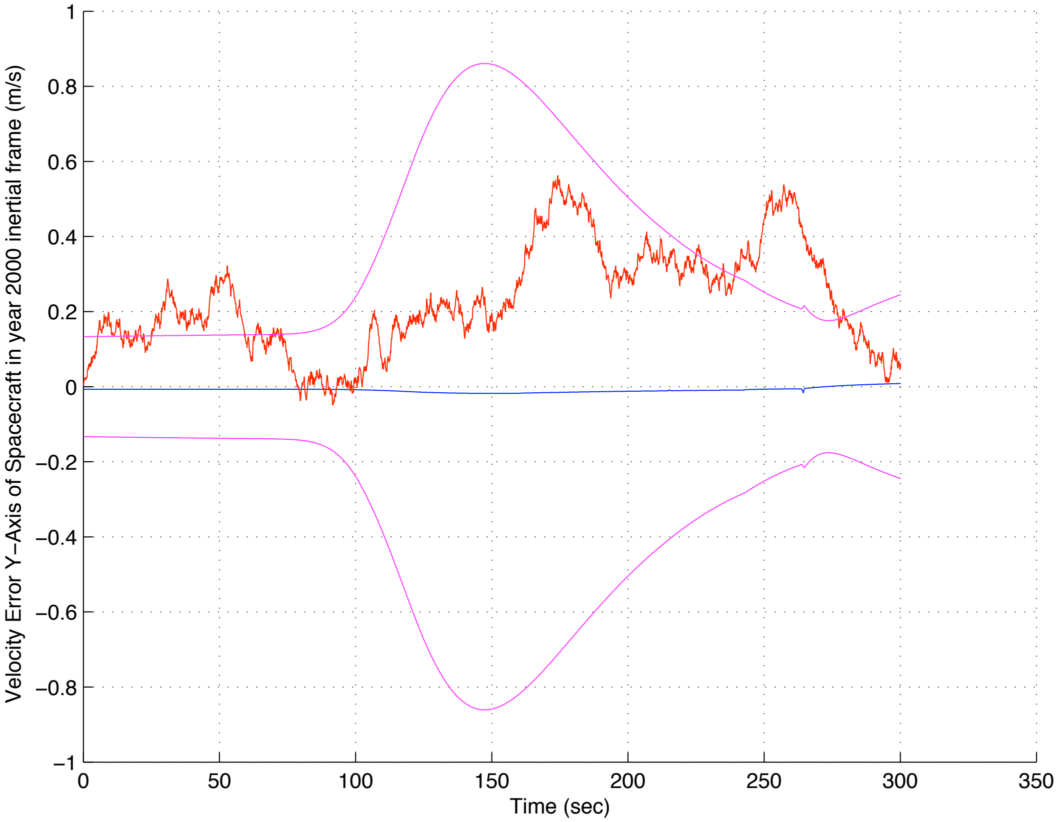

Figure 5.14: Monte Carlo average Velocity Error for Medium Noise year 2000 y.

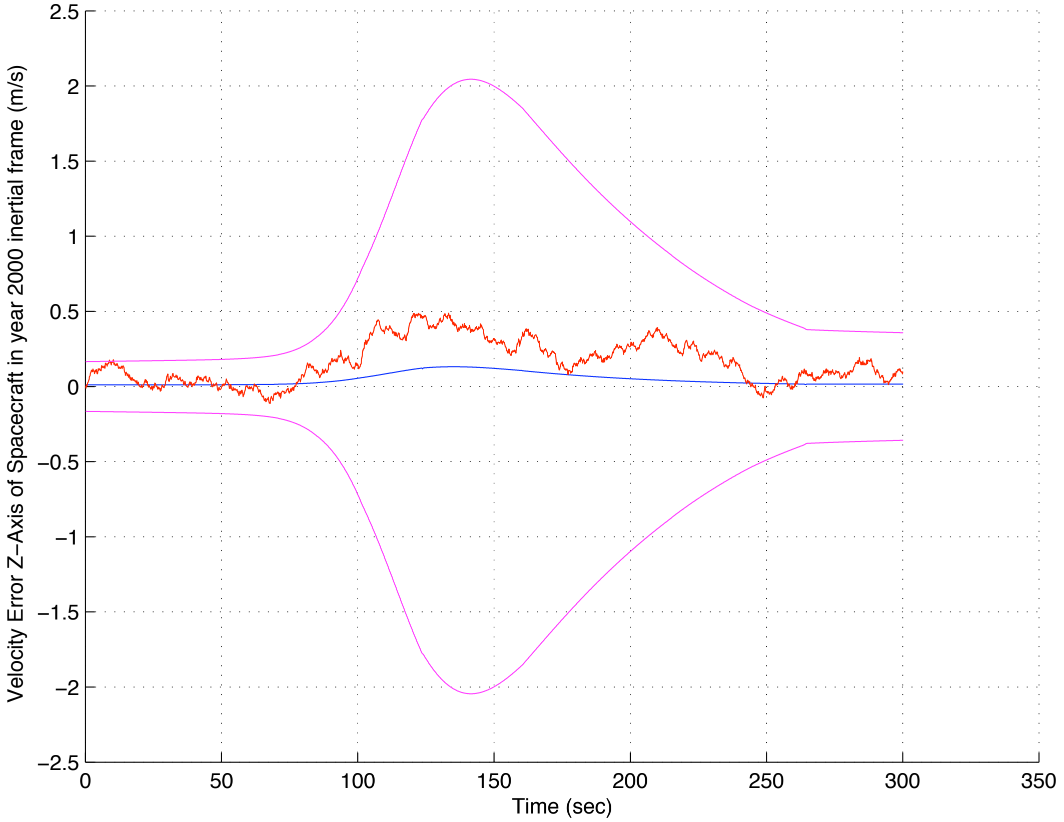

Figure 5.15: Monte Carlo average Velocity Error for Medium Noise year 2000 z.

5.1.2.3 High Noise Environment

In the presence of high noise, the average Monte Carlo estimate is shown in Fig. (5.16) through Fig. (5.18).

The result for the filter estimate here is a strong indication of the high quality of the model-based filter navigation concept. The *Monte Carlo* analysis shows that the filter estimate matches the truth well while no use can be made of the dead-reckoning solution is up to several orders of magnitude less accurate, as seen in Fig. (5.17).

For Mars EDL missions, the SPKF as a navigation tool can provide knowledge of the spacecraft state during the hypersonic phase of the entry and enable active guidance.

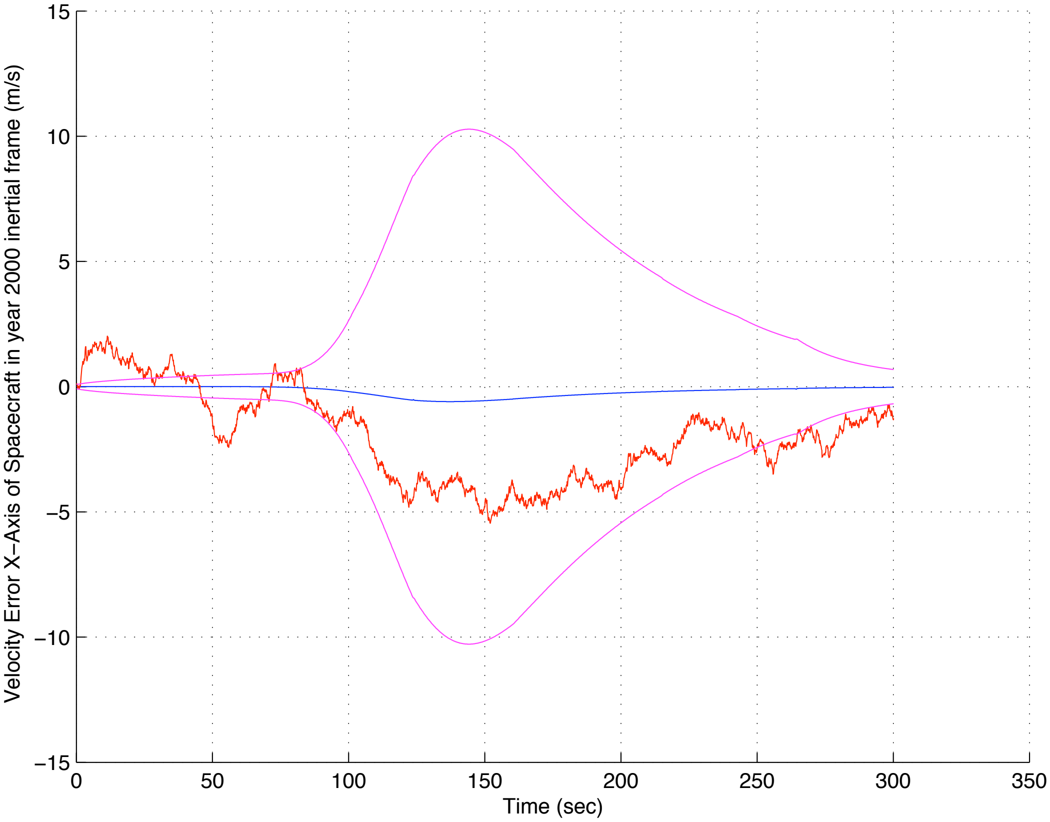

Figure 5.16: Monte Carlo average Velocity Error for High Noise year 2000 x.

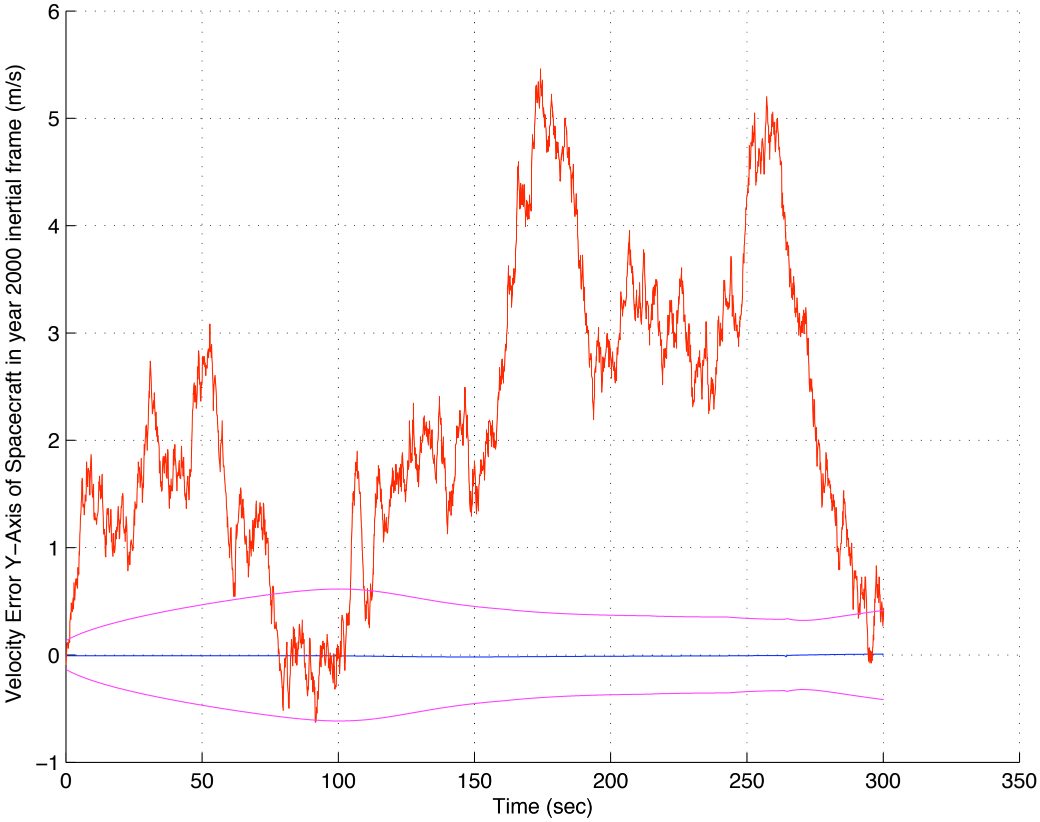

Figure 5.17: Monte Carlo average Velocity Error for High Noise year 2000 y.

Figure 5.18: Monte Carlo average Velocity Error for High Noise year 2000 z.

5.2 Second Entry Scenario

5.2.1 Filter Performance

Also the filter performance is considered for the the second entry mission. Again the same data is used to dead reckon the state and compare the results.

5.2.1.1 High Noise Environment

Using the filter with the measurement history with noise gives an estimate of the spacecraft velocity as shown in the plots here. Fig. (5.19) through Fig. (5.21) show the velocity estimates.

For one particular dispersed filter run the velocity estimate is shown. Again, the filter-based estimate clearly outperforms the dead reckoned solution. Beginning at cruise stage separation, the dead-reckoning process slowly leaves the true trajectory while the filter is following the trajectory very well until the atmosphere increases the uncertainty. At this point the filter shows some affect from uncertainties during entry, however the estimate continuously stays close to the truth and well within the covariance estimate. The relatively larger covariance in the filter estimate between the first entry scenario and the second entry scenario is due to the fact that the simulation tool has been developed for the the first entry scenario. The second entry scenario begins at cruise state separation and uses the same simulation tool which leads to higher uncertainties in the environment when the atmosphere begins to exert dynamic forces onto the spacecraft during EDL. It is recommended that the

SPKF be applied to the navigation purpose during the MSL mission.

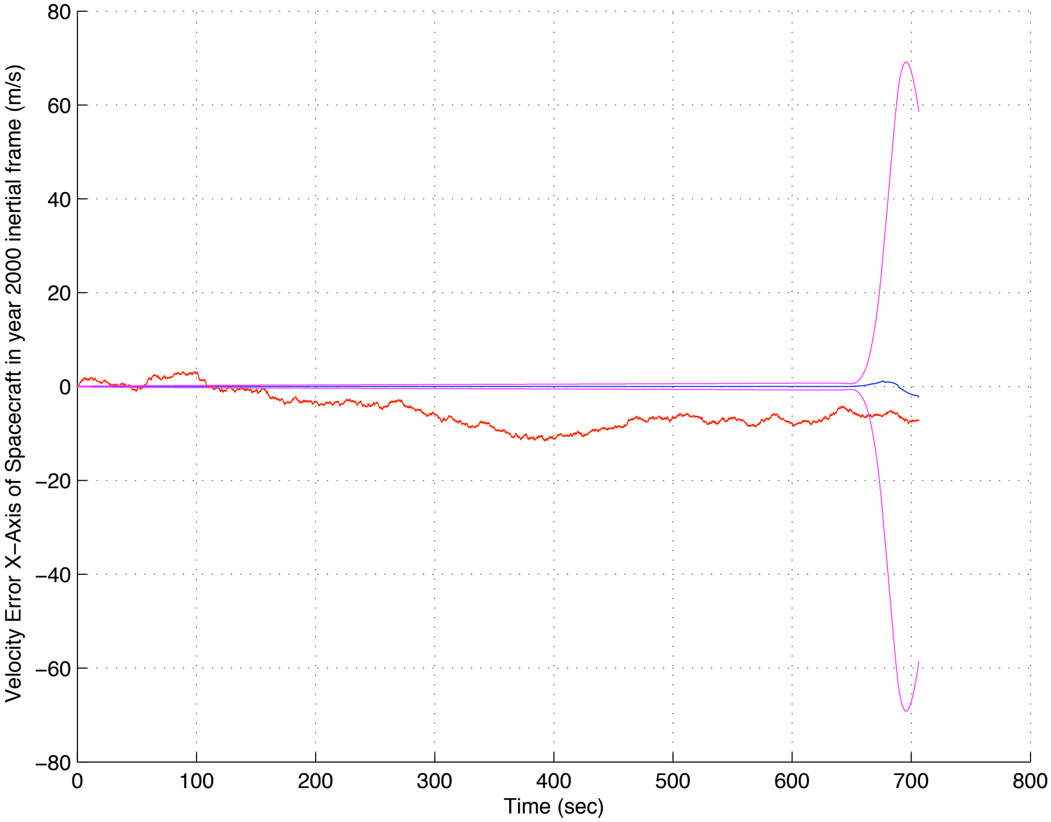

Figure 5.19: MSL SPKF Velocity Error for High Noise year 2000 x.

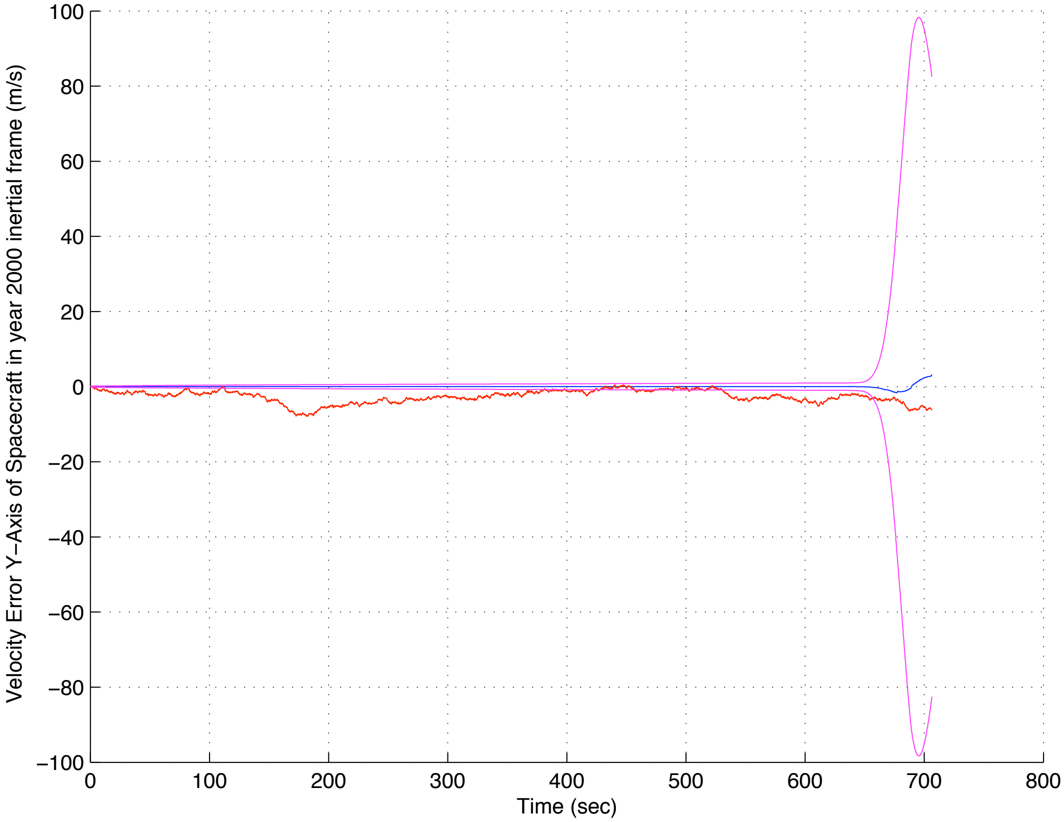

Figure 5.20: MSL SPKF Velocity Error for High Noise year 2000 y.

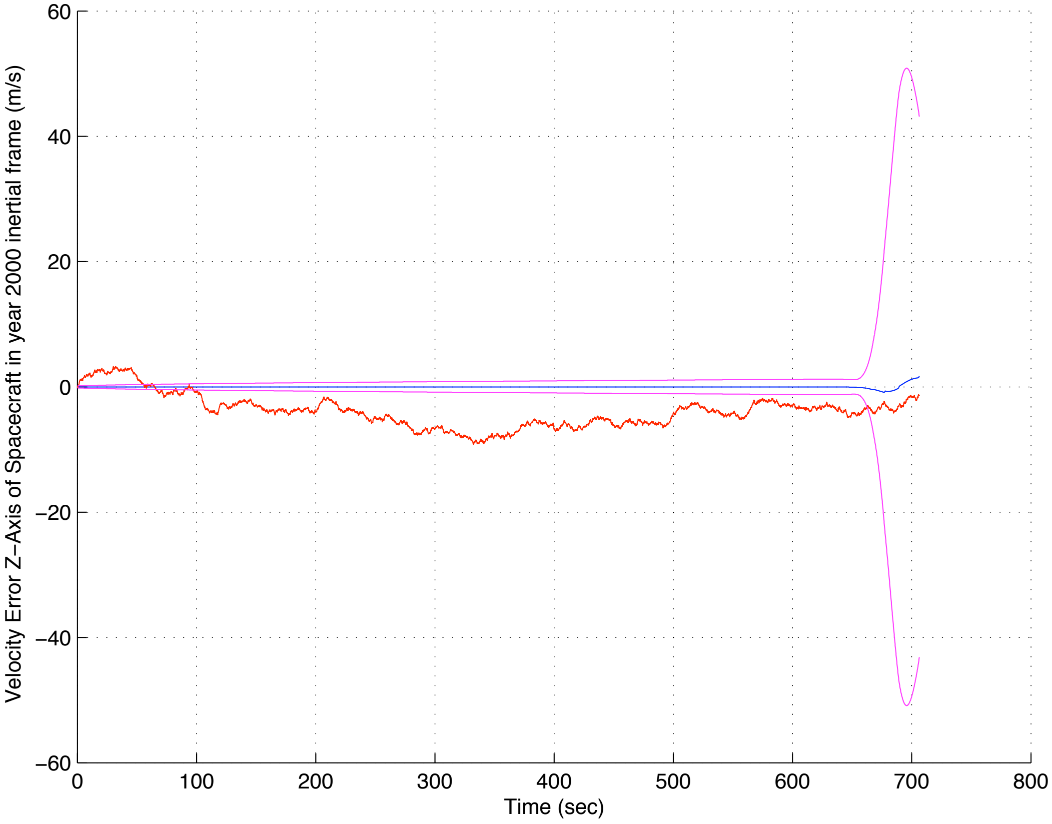

Figure 5.21: MSL SPKF Velocity Error for High Noise year 2000 z.

5.3 Autonomous Parameter Identification

Another interesting possibility that the SPKF offers is to determine environmental parameters of the system based on available measurements. To investigate this possibility, a bank of 11 filters was implemented of which one was the operational filter and the other 10 had different parameters with respect to the atmospheric density. The filter is set to start out at an incorrect setting for the atmospheric model and used to identify the proper parameter.

Filter 0 is the operational filter and Filter 6 is the filter with the correct atmospheric model. Filters 5 through 1 contain atmospheres that are of lower density and Filters 7 through 11 represent atmospheres with higher density models.

5.3.1 Low Noise Environment

In the presence of low noise, Fig. (5.22) shows the filter selected at each time step. Fig. (5.23) through Fig. (5.25) show the velocity estimate of the operational filter and dead-reckoned states combined with the self-tuned filter and the manually-tuned filter. The solid purple line is the state covariance of the self-tuned filter while the corresponding dashed line is the state covariance of the manually tuned filter.

While initially the atmospheric effect is very low, the filter still manages to adapt and pick the correct model to use even before the atmospheric drag becomes significant. Once identified the filter continuously remains with the proper atmospheric model.

The quality of the estimate is equal to the manually-tuned filter when compared to the dead-reckoned trajectory in the presence of low noise. However, the covariance of the autonomously tuned filter is larger than when tuned manually. The plots show that the manually tuned state estimate practically coincides with the autonomously tuned filter estimate. At times the autonomous model selection is resulting in a better estimate than the fixed model estimate, though the uncertainty in the state is larger for the autonomous model selection. As in the analysis for the filter with a fixed model, the autonomous parameter selection works well and does provide a good state estimate. However the dead-reckoning in the presence of low noise and excellent knowledge of the initial conditions also is giving a valid estimate.

The closely matching results for the fixed model estimate and the autonomous model selection indicate that the filter bank is suitable to be used in the navigation architecture.

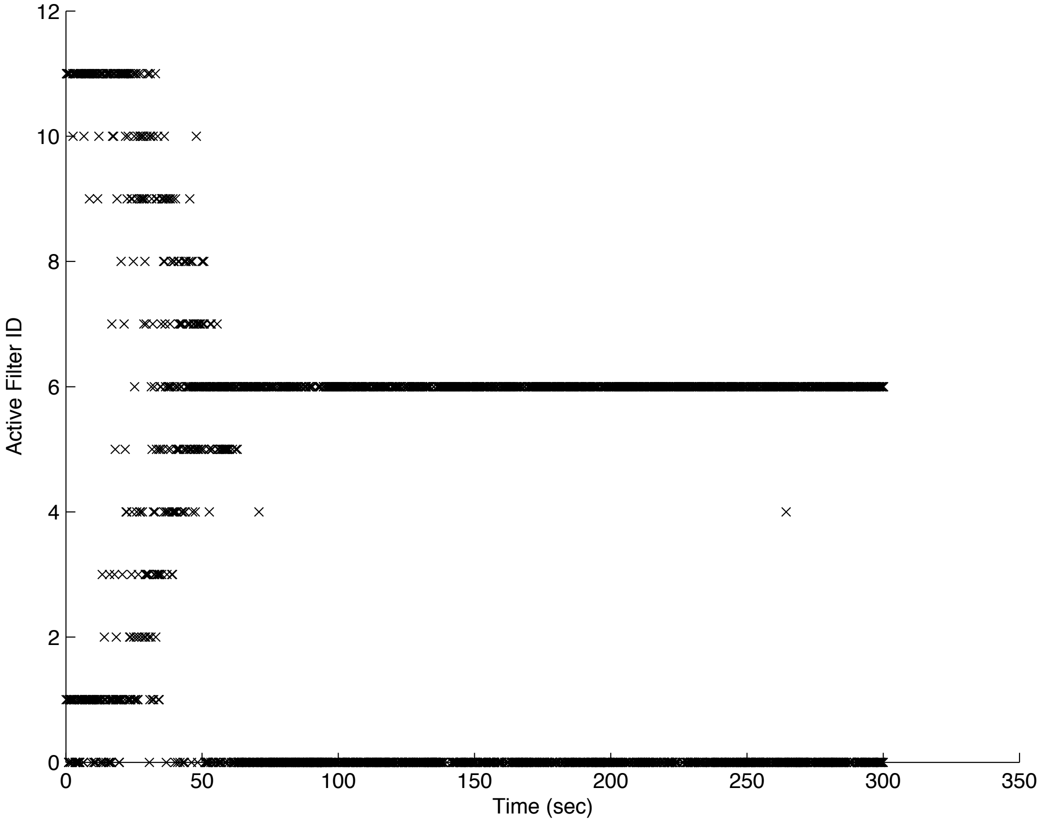

Figure 5.22: Selection of Atmospheric Model in Low Noise.

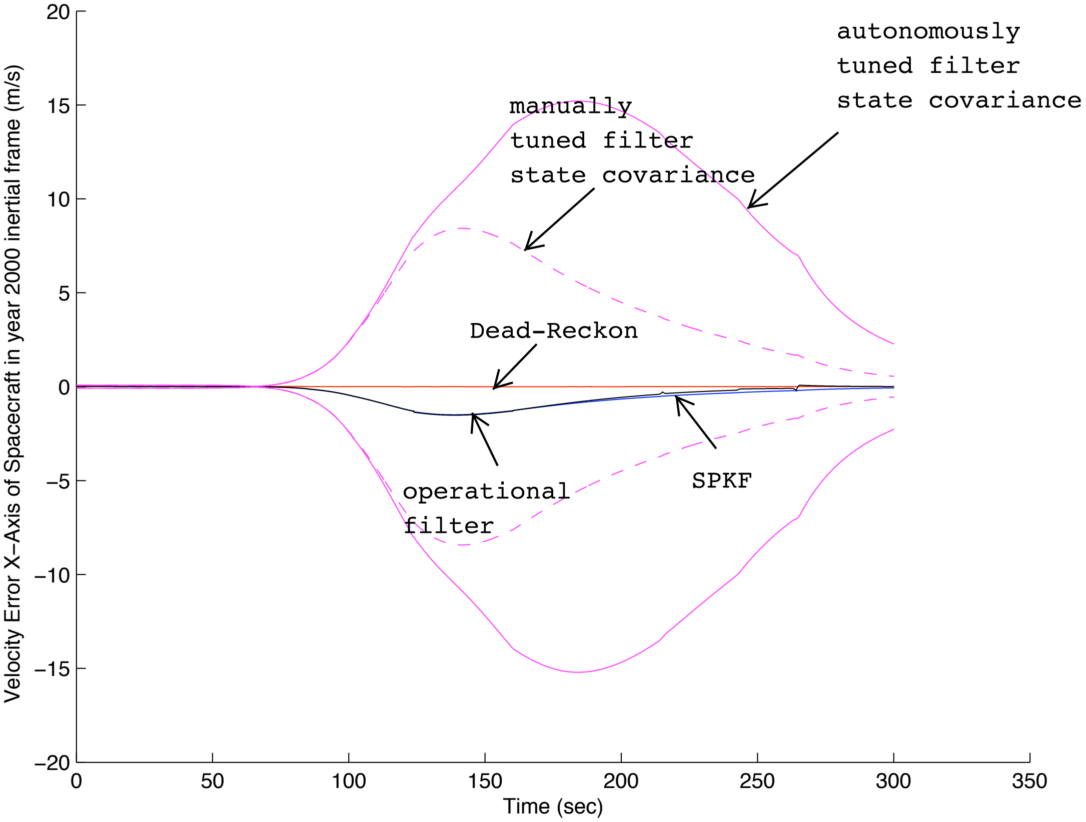

Figure 5.23: Velocity Error for Low Noise year 2000 x.

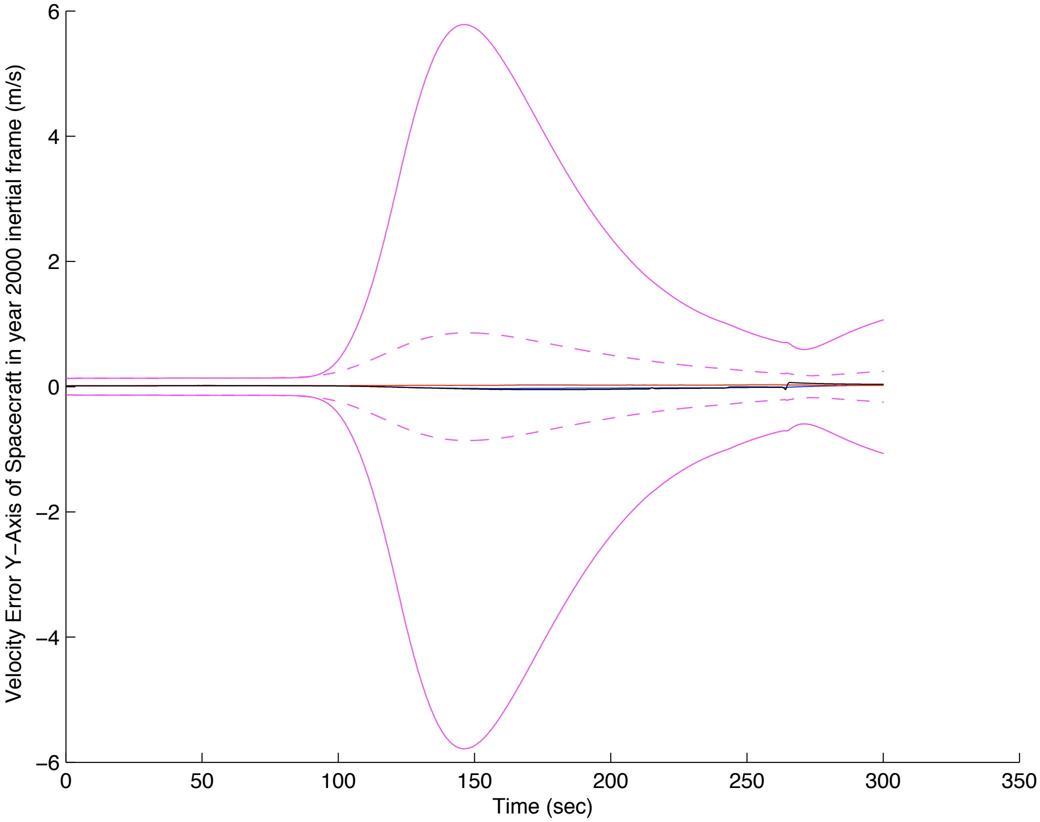

Figure 5.24: Velocity Error for Low Noise year 2000 y.

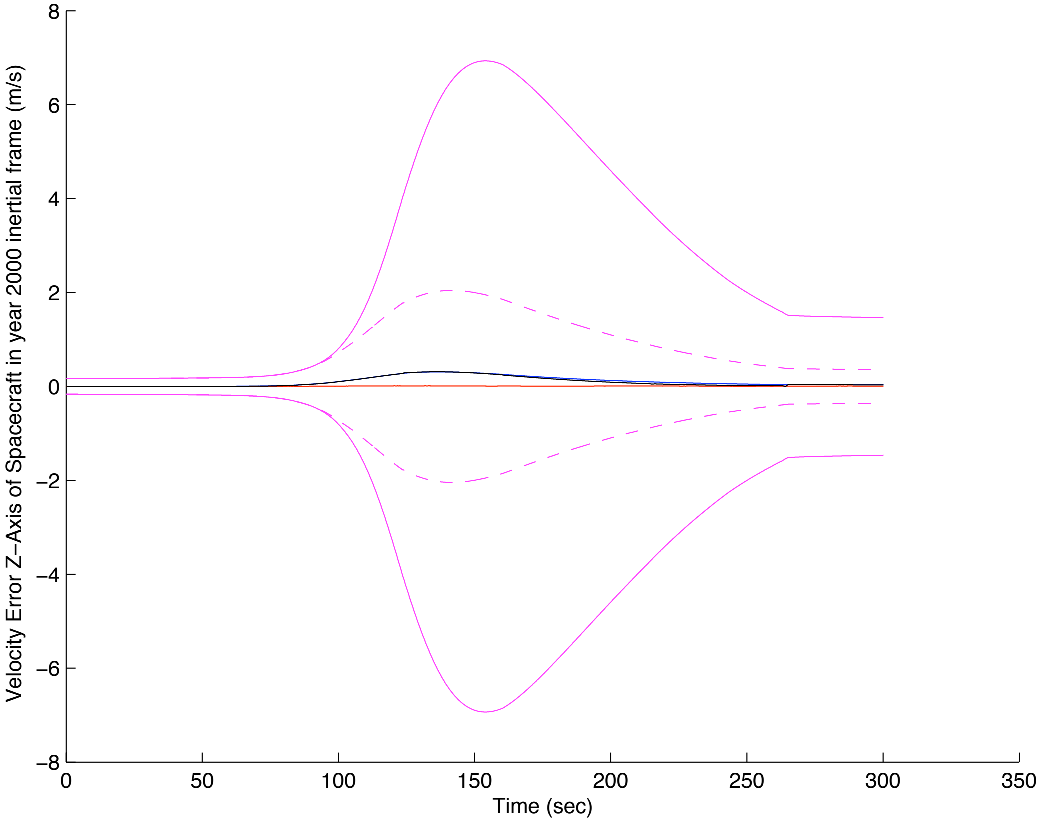

Figure 5.25: Velocity Error for Low Noise year 2000 z.

5.3.2 Medium Noise Environment

In the presence of medium noise, Fig. (5.26) shows the filter selected at each time step. Fig. (5.27) through Fig. (5.29) show the velocity estimate of the operational filter and dead reckoned state combined with the self tuned filter and the manually tuned filter.

While initially the atmospheric effect is very low, the filter still manages to adapt and pick the correct model to use even before the atmospheric drag becomes significant. Once identified the filter continuously remains with the proper atmospheric model.

The quality of the estimate is equal to the manually tuned filter when compared to the dead reckoned trajectory in the presence of medium noise. However the covariance of the autonomously tuned filter is larger than when tuned manually. Though again it can be seen that the autonomously tuned filter does provide a better estimate than the dead reckoning of the data in the presence of medium noise.

This is again analogous to the fixed model results. It is advantageous to use the filter based estimate over the dead-reckoned estimate as the actual state is matched well with the filter as opposed to the dead-reckoning of the IMU. Again the uncertainty in the autonomously selected environmental model is larger than the result for the fixed model.

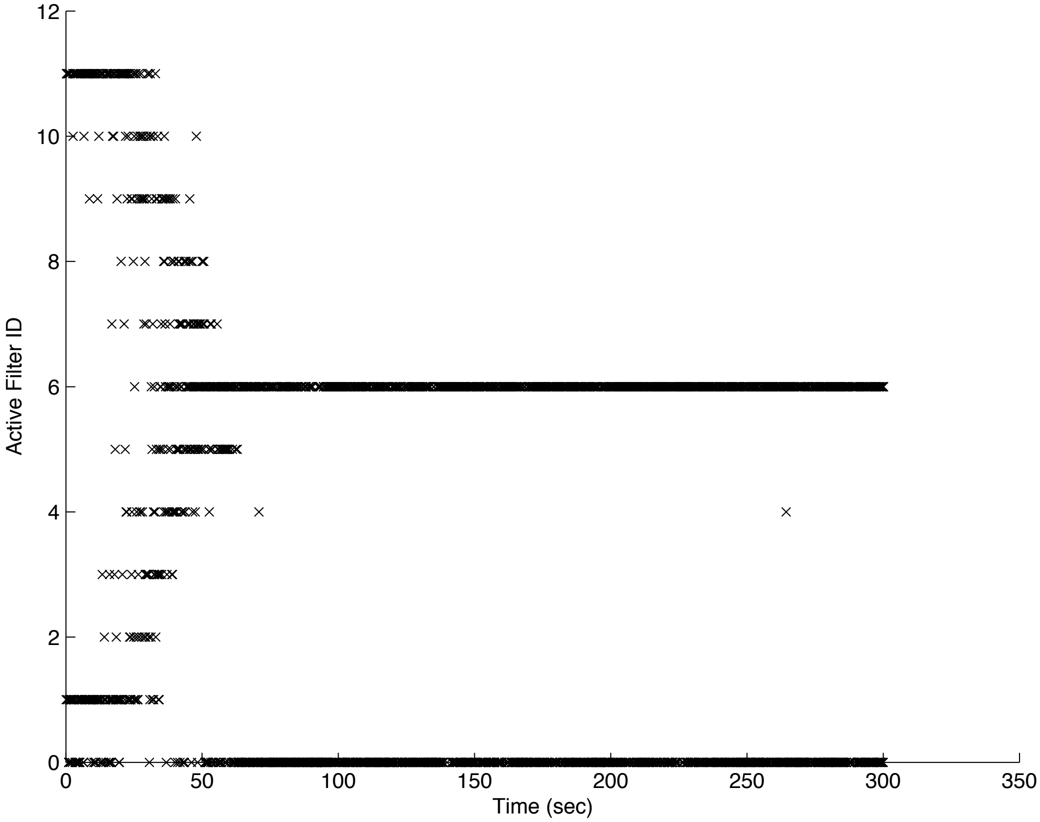

Figure 5.26: Selection of Atmospheric Model in Medium Noise.

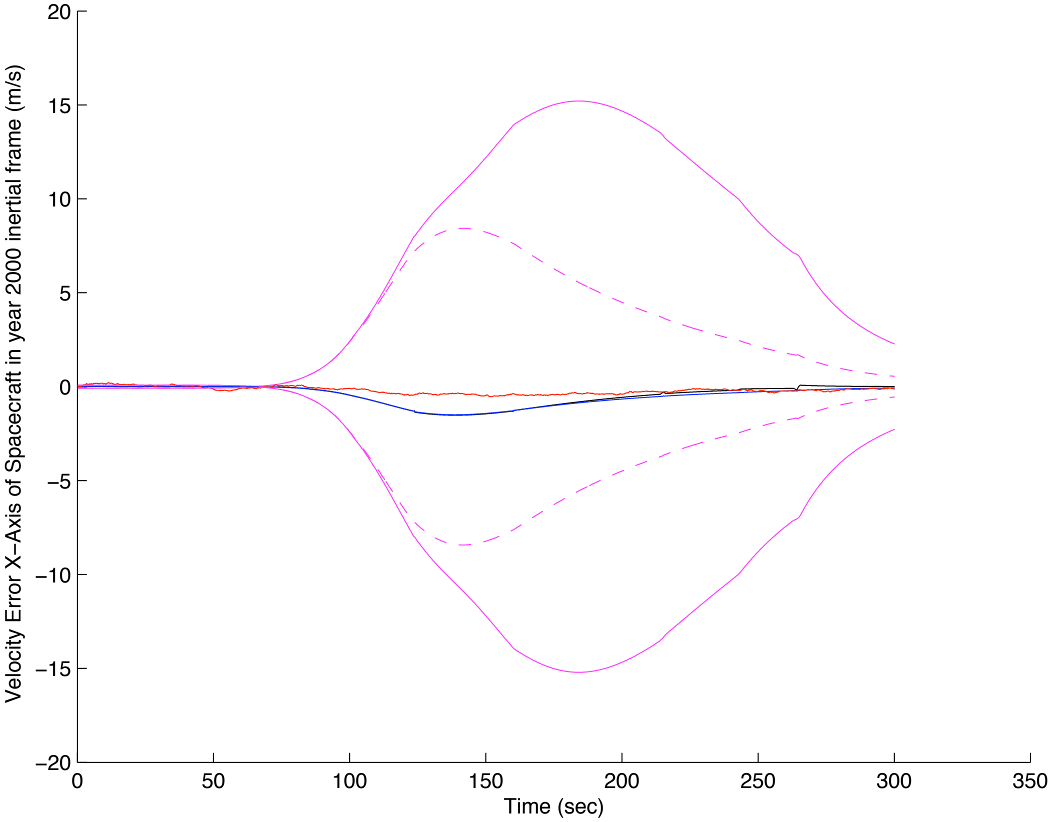

Figure 5.27: Velocity Error for Medium Noise year 2000 x.

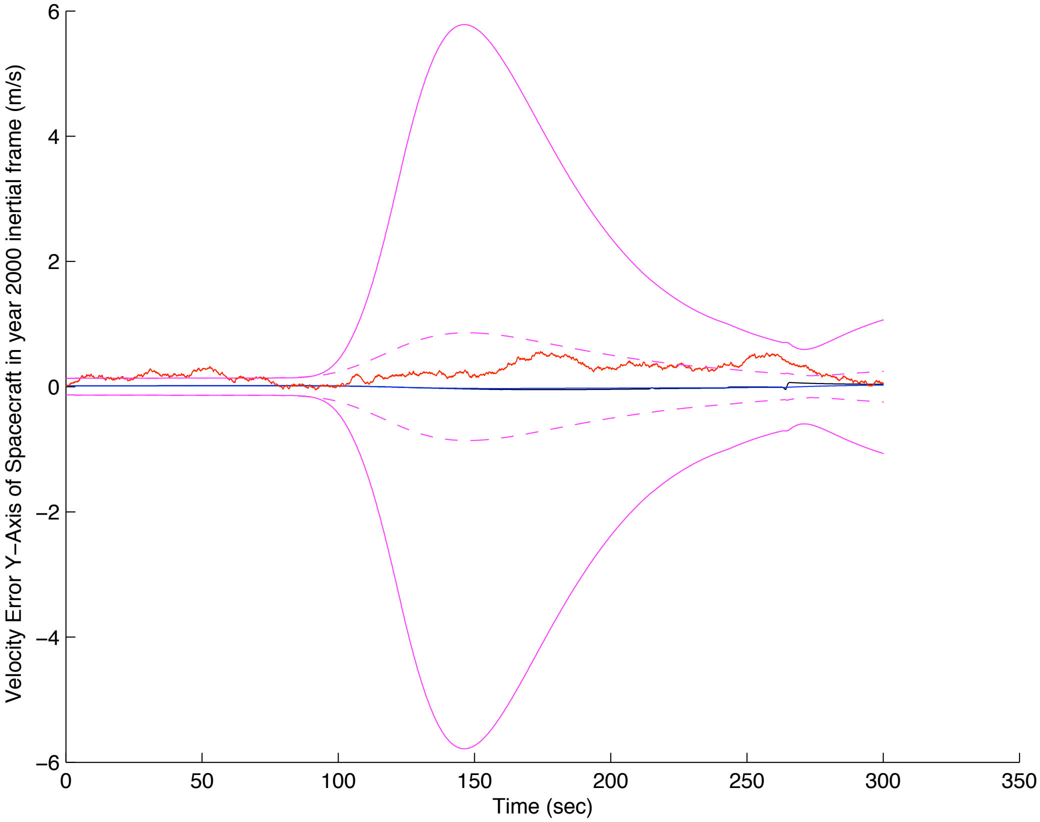

Figure 5.28: Velocity Error for Medium Noise year 2000 y.

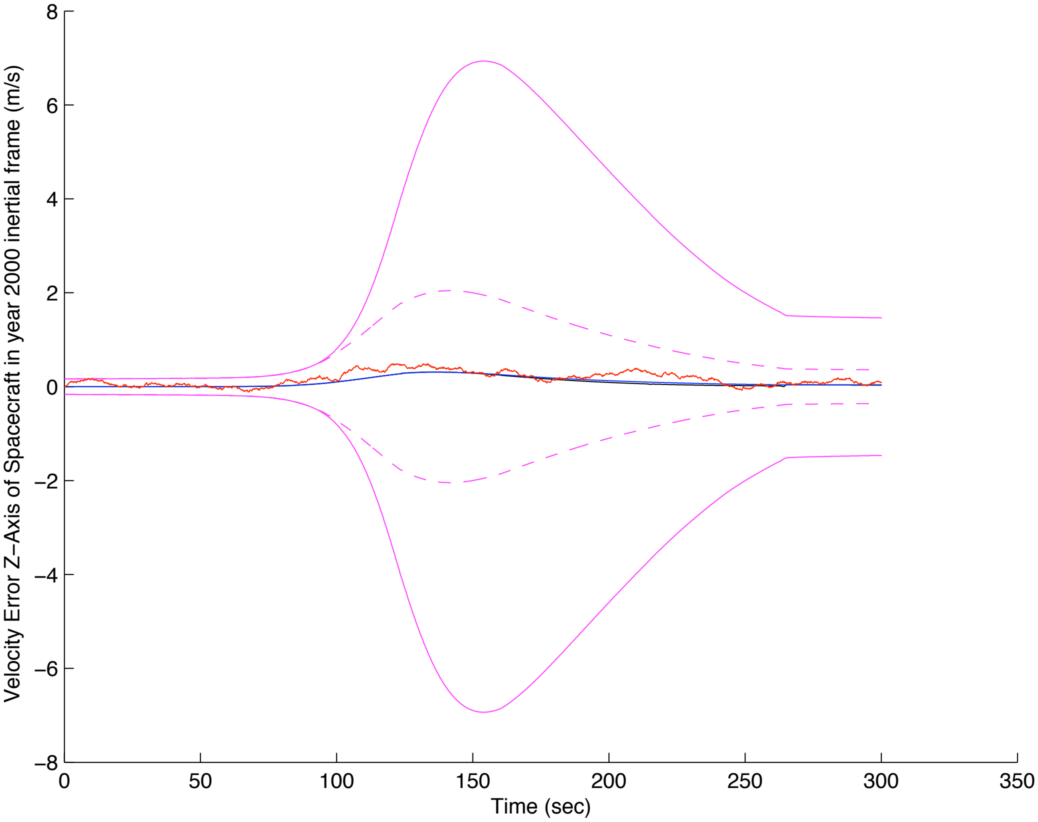

Figure 5.29: Velocity Error for Medium Noise year 2000 z.

5.3.3 High Noise Environment

For high noise the selection of the environmental model proved more difficult. This case is illustrated in Fig. (5.30) which shows the filter selected at each time step. Fig. (5.31) through Fig. (5.33) show the velocity estimate of the operational filter and dead reckoned state combined with the self tuned filter and the manually tuned filter.

With the high noise environment, the filter is able to identify the atmosphere. The filters that are mostly selected throughout the simulation are within 2 % of the actual atmosphere and the size of the noise caused the filter bank to switch between the filters within 5 % continuously. Towards the end of the simulation when the dynamics reduce and the measurements are of smaller magnitude, the noise begins to have a notable effect. The noise causes the filter bank to pick models of higher and lower density than the actual environment.

The fact that the correct atmospheric model remains unidentified causes the autonomously filter to perform less accurate than the fixed model implementation. Combined with the estimate of the uncertainty however, the filter still performs acceptable. While it fails to operate to the quality of a correctly modeled environment, the state estimate remains within the estimated uncertainty range and also still give better results than the dead-reckoning process.

The quality of the estimate of the autonomous model selection confirms the trends established for the manually tuned filter when compared to the dead reckoned trajectory in the presence of noise. However, the covariance of the autonomously tuned filter is larger than when tuned manually. Though again it can be seen that the autonomously tuned filter does provide a better estimate than the dead-reckoning of the data in the presence of noise. As the filter picks the atmosphere of too high and too low density towards the end of the simulation, the overall estimate still performs well.

This single run for the high noise environment confirms the ability of the filter bank to give a valid state estimate and matching covariance as anticipated for the navigation concept proposed in this work.

The filter bank performs well within limits in the range of environmental deviation. The actual environment must be reasonably close to the models included in the filter. An experiment has been performed with measurements representing an environment significantly deviating from the nominal and far from the models implemented in the filter bank. The filter for a good period identifies the correct environment but once thrown off is not capable to recover from the false selection of parameters and fails to identify the true environment.

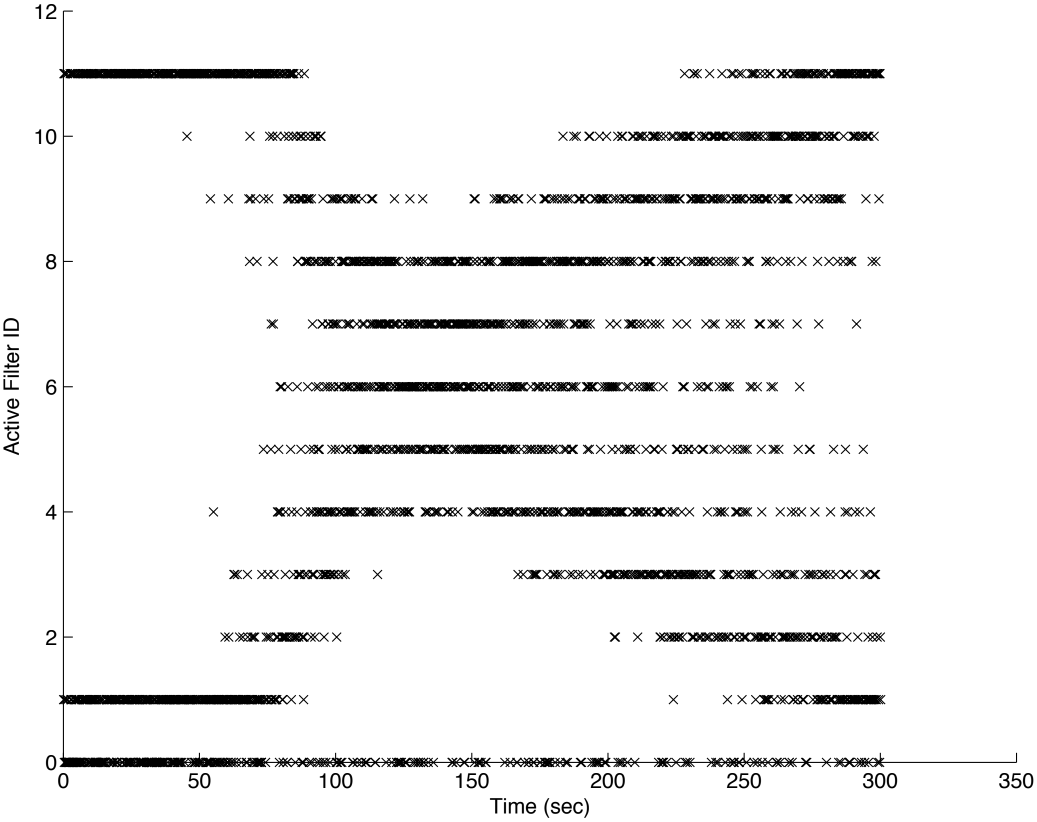

Figure 5.30: Selection of Atmospheric Model in High Noise.

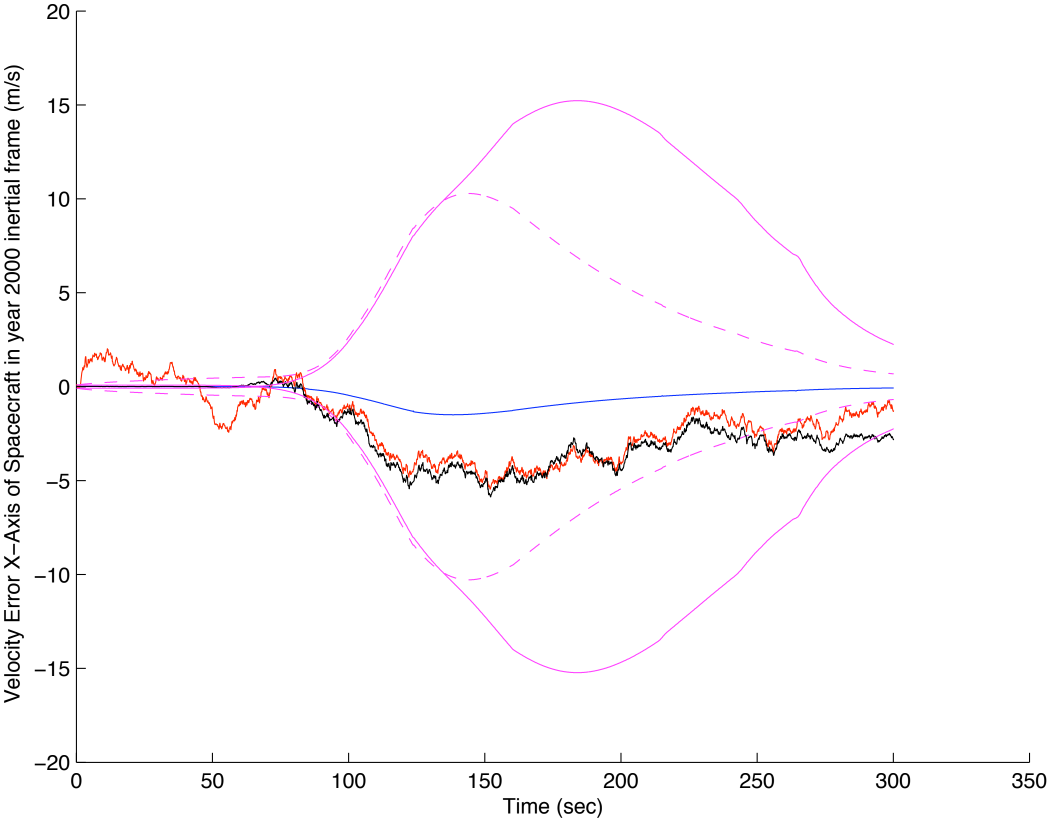

Figure 5.31: Velocity Error for High Noise year 2000 x.

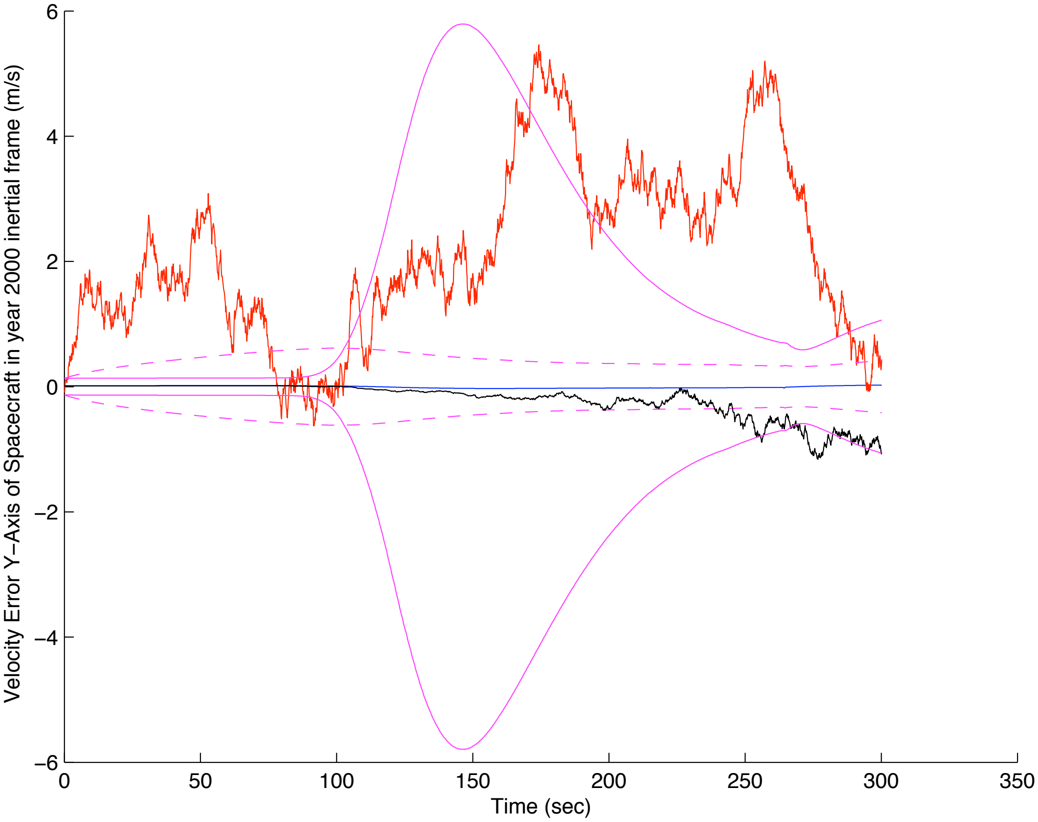

Figure 5.32: Velocity Error for High Noise year 2000 y.

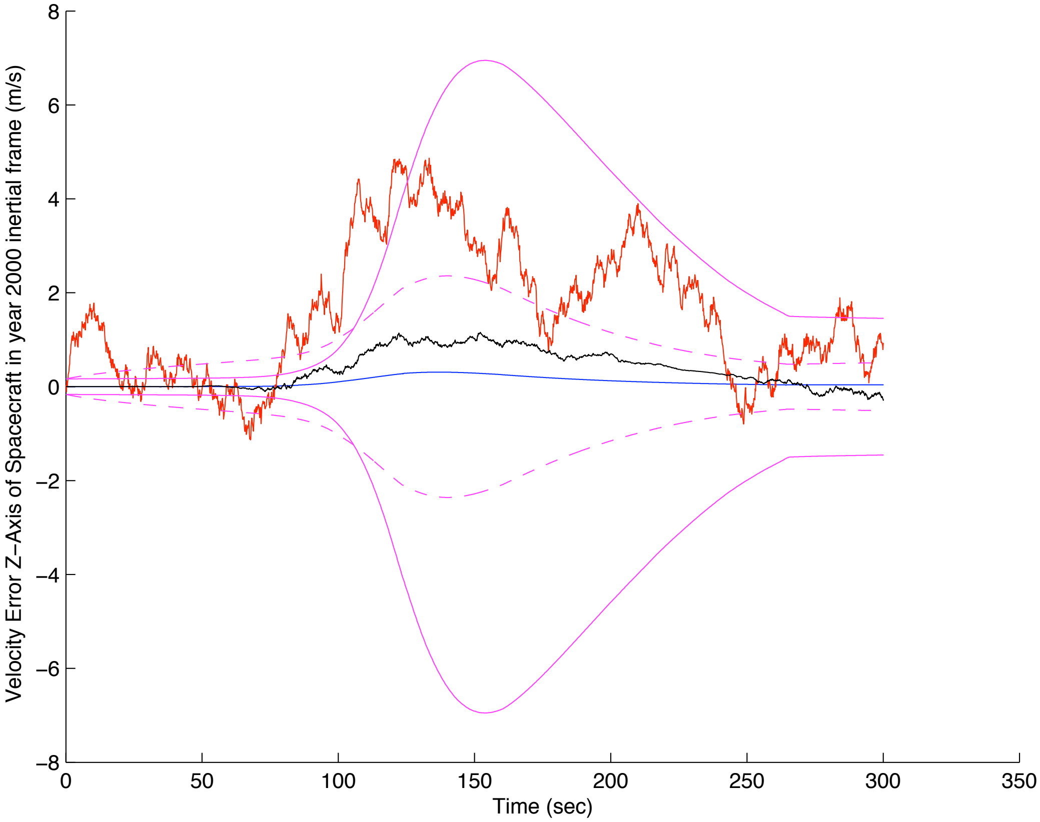

Figure 5.33: Velocity Error for High Noise year 2000 z.

Chapter 6

Hardware Implementation

An integral part of this work is the implementation of the navigation concept on the Electra Programmable Radio to validate it for use on actual flight-hardware. It is important to demonstrate that the navigation architecture proposed here can be used in an actual in-flight application on existing space infrastructure. In order to achieve this, the constraints imposed by the flight-system must be identified and the navigation concept be adjusted as necessary. An understanding of the hardware is essential. Also, an understanding of the tools in existence for programming the platform is required.

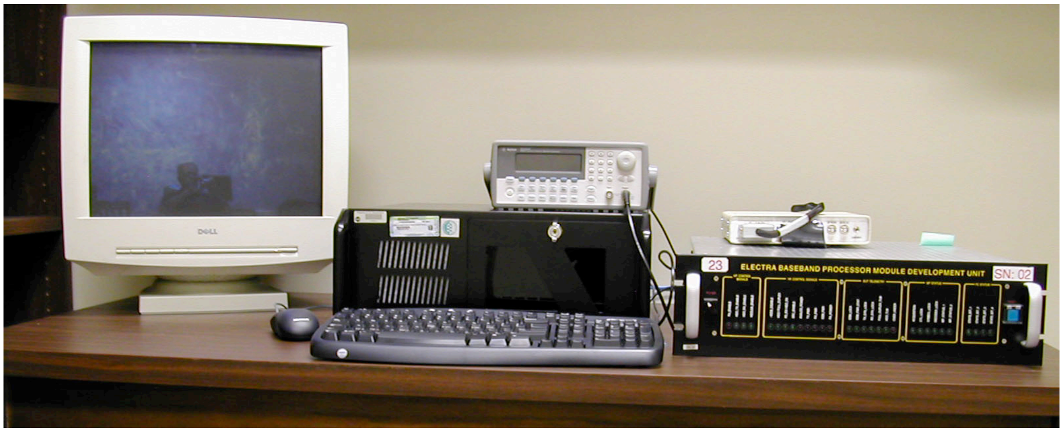

Figure 6.1: The Electra Programmable Radio in the UT Navigation Laboratory.

6.1 The Electra Programmable Radio

In order to build communication infrastructure at Mars, the Electra programmable radio was developed by NASA to serve as a data node. Electra is designed to improve communication capabilities between spacecraft at Mars and the command centers on Earth. With Electra being part of every mission to Mars beginning with Mars Reconnaissance Orbiter (MRO), it will improve data transfer capabilities between Earth and Mars.

An additional capability that Electra offers is as a navigational beacon. With an accurate knowledge of the state of a spacecraft carrying the Electra in Mars orbit, incoming spacecraft can use Electra as a 1- or 2-way doppler beacon to more precisely determine the state of spacecraft arriving at Mars. The same idea applies to rovers on the surface of the planet. In this sense, Electra represents an integral element of the communication and navigation infrastructure that will be continuously improved with new spacecraft arriving at Mars.

As the baseline design of the Electra platform offers excess computational capabilities, it can be considered as a component of an autonomous on-board GN&C system to further improve the navigation system for missions to Mars.

It is envisioned that Electra can be used to provide a real-time estimate of the spacecraft state to the on-board guidance and control system. In order to achieve this, sensory input (such as IMU data) must become available to the Electra that can then be processed and a state estimate be provided by the platform. This idea was implemented as part of this work.

In order to develop software for the Electra hardware, a variety of constraints had to be considered. The concepts were evaluated through standalone C/C++ based simulation tools built for the purpose of off-line testing. These concepts were then carried over into the Electra emulator to verify their functionality on the hardware. Upon successful completion of the off-line testing, the codes were ported to the hardware and verified on the Electra platform.

6.2 Electra Emulator

The Electra emulator is available on the computing facilities at UT. It was provided by JPL for the purpose of software development and validation for the Electra hardware. The emulator allows the use of any connected computer to test code on an Electra equivalent basis. Code successfully implemented on the emulator has commonly been ported to the hardware and executed successfully with minor adjustments for most applications. The major additional restrictions for successful execution on the hardware are memory management and size restrictions on the executable file.

6.3 Electra Hardware

Electra is a software in-flight reprogrammable radio which is carried by MRO, MSL, and future orbiter/ lander missions to Mars. It is currently capable of collecting integrated Doppler data. In the nominal configuration, Electra has spare processing/memory capability suitable for higher level functionality. The proposal is to access the spare computational capabilities and memory for navigation. There is V7 Sparc processor on Electra running at 24 MHz. About 2 Mb of runtime memory are available for the executable file, while 13 Mb file storage can be utilized. The proposal is to port IMU data to Electra, in addition to the available doppler data.

With Electra becoming part of Mars communication and navigation infrastructure, it is an ideal platform to demonstrate real-time embedded navigation technologies without compromising the flight system baseline GN&C system.

6.3.1 Hardware Constraints

In the process of the implementation, various limitations of the Electra hardware had a major impact on the development of the navigation filter. This was expected. These constraints which impose demands on the structure of the software implementation come from a variety of sources in the hardware design.

The restrictions that had an impact of the development were the Electra operating system, the space available for runtime memory, the method of accessing external measurements, the structure of the memory addressing and the processor speed.

6.3.1.1 Electra operating system

As the use of Electra as a navigation tool is still in the development stage, the operating system providing IO functions is still far from completion for the requirements of a navigation system. Access to sensor data is not available, neither is there a direct way to display memory content in an efficient format or port it as an output from Electra. Various modifications to the software were necessary to implement the concept presented in this work.

6.3.1.2 Runtime Memory

The limit of 2Mb available for the size of the executable file on Electra required careful attention to efficient programming for the navigation filter. This was of particular importance as the creation of any executable code had various libraries that had to be included for proper execution. This preexisting structure required roughly 1.5 Mb of space which significantly limited the room available for the navigation algorithm to be implemented.

While the Electra emulator did not have to observe this size limit for the executable file, the implementation on the emulator was achieved quickly while a variety of modifications were necessary to fit the algorithm onto the hardware.

The major difference in the code that was used on the emulator versus the code that could be executed on the hardware is the way the IMU measurements are stored and accessed during time of execution.

6.3.1.3 External Measurements

In order to process IMU data in the navigation filter, the data had to be available to the hardware at the time of the filter algorithm measurement update. As no function is implemented as part of the current Electra OS that grabs external sensor data, this had to be accomplished in a different way.

As the measurement data for a complete filter run could not fit as part of the executable file, a method had to be developed to have the data available on Electra without it becoming part of the executable file. In order to achieve this, a LabView interface script was developed that placed the IMU data in a memory section which was not used by Electra during execution.

6.3.1.4 Memory Management

A further obstacle that had to be resolved was the way data is placed in the Electra memory. The memory addressing on Electra displays 'holes' or non-existing memory addresses. When reading IMU data from memory the various bytes that constitute a double variable had to be read from the scattered memory locations and recombined into a variable that could be properly read as a double variable.

The memory that is available for use after the executable file has been placed on the hardware begins with the address `0x 14 00 00 00`. This sector of memory runs though `0x 14 00 0F F3` and then jumps to `0x 14 00 10 00`. This continues in the same manner throughout the memory. The 2nd sector runs through `0x 14 00 1F F3` and then jumps to `0x 14 00 20 00`. Within the memory sectors the data is organized in a way that the Electra interface places data in every fourth block of 4 bytes.

As a double variable consists of 8 bytes, the content of two blocks of 4 bytes contain a double variable. As a result, in order to read data from the Electra memory these two blocks of 4 byte each need to be read and combined in a continuous memory location from where the double variable is read and processed by the software.

The implementation was accomplished using eight pointers to the memory location of the double variable and copying the content of the appropriate bytes from the Electra memory into this continuous memory of the double variable. For the first double variable this means that the data from addresses `0x 14 00 00 00`, `0x 14 00 00 01`, `0x 14 00 00 02`, `0x 14 00 00 03`, `0x 14 00 00 0C`, `0x 14 00 00 0D`, `0x 14 00 00 0E` and `0x 14 00 00 0F` are recombined as the first double variable. Attention must be paid at the end of each sector of memory as `0x 14 00 0F F3` is followed by `0x 14 00 10 00`, but at the address `0x 14 00 10 00` data is stored and the logic continues as in the block beginning at `0x 14 00 00 00`.

6.3.1.5 Processor Speed

As the navigation concept was successfully implemented as a standalone code and modified to properly run on the Electra hardware, it became clear that the processor speed on Electra does not allow the calculation of a state estimate in the time required to have it available during the actual EDL phase of the mission. With an increased processor speed the concept should however become a part of future navigation system implementation. The processor on the new Electra programmable radios has already been improved by a factor of 4. This is expected to continue to improve.

6.4 Hardware Interface

In order to interface with the Electra hardware successfully, a Labview interface (known as virtual instrument or VI) was developed that commands the hardware. The Labview VI initialized Electra, transfers files to the hardware, and observes the Electra state during simulated runs.

The Labview VI interface was provided by JPL [21]. Minor adjustments in the interface had to be made in the process of resolving the memory access issues encountered.

6.5 Hardware Real Time Simulation

With all the limiting factors taken into account the navigation concept was hosted on the hardware and tested. Upon providing Electra with all required software, simulations were performed on the actual hardware. As the computational capabilities of Electra are limited and smaller than the laboratories resources, hardware runs are performed for validation of stand-alone and emulator simulations only. Execution speed is a limiting factor on the hardware. It was determined that Electra results were identical to those obtained on the emulator.

6.6 Electra Results

With the implemented filter architecture, the applicability of the concept for the task of hypersonic entry navigation was evaluated. As the off-line *Monte Carlo* analysis proved, the estimation of spacecraft state is performed successfully by the SPKF.

The mean plots shown in Section 5.1.2 for the *Monte Carlo* runs display the filter ability to estimate the spacecraft velocity states successfully with the sensor input available. The position states are found in a direct correlation to the velocity estimate which corresponds to the expectation as the position is the integral of the velocity.

The SPKF is operational on the Electra hardware and is suitable for the task of state estimation for Mars EDL navigation. The position estimate is a direct function of the velocity knowledge as the information available from the IMU only provides data in the velocity channel. The velocity states of the spacecraft are estimated well and clearly outperform the time update only.

The majority of the dynamics take place in the inertial X axis in the year 2000 Mars centered frame. Even at the time of the highest accelerations encountered, the predicted spacecraft state is very close to the truth. The behavior in the well behaved Y and Z axis follows the truth easily.

It is interesting to observe the history of the sigma points during the entry. Observation of the sigma points in the velocity plots shows how the estimate stays well with the truth and has the highest deviation away from the truth during the time of the highest activity in the system dynamics. This takes place at the point where the atmosphere of Mars begins to take effect on the spacecraft and continues to be the major force on the spacecraft until it has been significantly decelerated. After this period the changes in the state become smaller and the estimate returns closer to the truth. With the dynamics slowing down, the certainty in the estimated state grows again. The sigma point plots show how the sigma point histories govern the covariance values.

As for the simple time update of the dynamic system, a better result is achieved by implementing the SPKF over dead-reckoning. It has been demonstrated that the use of the SPKF for the navigation purpose is an improvement of the knowledge of the spacecraft state. It can be reasonably expected that further improvements can will be seen when additional measurements are utilized as part of the measurement update process.

The implementation of the navigation architecture on the Electra hardware has been successfully performed. The results from the stand-alone filter, emulator and hardware implementation give the same results.

Chapter 7

Conclusions and Future Direction

For the first time a sigma point Kalman filter has been implemented for the task of autonomous Mars entry during the upper atmosphere hypersonic phase. The successful implementation for the hypersonic entry at a distant planet with an atmosphere points to the liklihood that the implementation for the remaining portions of the descent which display less complexity in their dynamics while providing more sensory input will be desirable.

7.1 Achieved State

This implementation of the SPKF as developed under this project represents a substantial improvement of traditional navigation methods. The benefit for the Electra program extend well beyond the scope of this thesis and have the potential to impact Mars spacecraft and rover navigation on a sustained basis. A well-tuned filter for a specific environment is capable of performing the required pinpoint precision navigation task. However in the presence of environmental uncertainties as often a concern for autonomous interplanetary lander missions, the use of a SPKF filter bank promises to be of benefit.

As demonstrated in this work, the filter bank is capable of identifying model parameters and adjusting the filter for the environment through which it is passing. With sufficient computational power a large filter bank addressing various uncertainties in the environment during EDL is expected to enable the navigation system to be robust to many different deviations.

The integration of a navigation capability onto the Electra programmable radio has been successfully performed and can be considered as an aid to spacecraft navigation for future planetary exploration missions. For the implementation of the filter bank on Electra additional processing speed must be made available on the hardware to achieve a flight-ready design. Currently the platform is computationally too slow to handle the amount of calculations necessary for the task.

7.2 Future Work

The concept demonstrated here for the hypersonic EDL segment can be applied to further mission phases that need to be subject to continued research for the SPKF applicability. As new sensors become available at the end of the hypersonic EDL phase at parachute deployment, the navigation accuracy for the remaining portion is expected to be high while the hypersonic phase is essential to arrive at conditions at which the remaining descent is able to reach to touchdown point on the surface of the planet.

It is a project of further interest to develop a computationally efficient implementation of the architecture proposed to make optimal use of the resources and arrive at a precision estimate for an autonomous mission.

Regarding the advancement of a adaptable filter bank, further work can be performed to validate the architecture's ability to simultaneously identify a variety of environmental parameters such as gravitational and atmospheric conditions as well as vehicle parameters.

With advances in processor speed and parallel computation it will become a viable possibility to test the filter bank during an actual mission. A first step could be the use of the LEON-2 or LEON-3 processor in the Electra hardware which is a Space-based core and higher processing speed.

The architecture should be used in an in-flight experiment to validate the concept. A mission suitable for this test is Mars Science Laboratory as it has all the required elements. The Electra hardware has already been demonstrated to have the ability to perform the tasks associated with the concept. It is therefore a recommendation to proceed with an in-flight test of the SPKF for EDL navigation.

Appendices

Appendix A

Trajectory Comparison

This Appendix contains plots which were used to compare the various trajectories used in this work. The difference between them in the relevant axes are plotted.

A.1 MER-A Trajectory Analysis

A.1.1 JPL MER-A vs. UT Simulation

The differences between the MER-A JPL provided trajectory and the UT simulation are shown in Fig. (A.1) through Fig. (A.6).

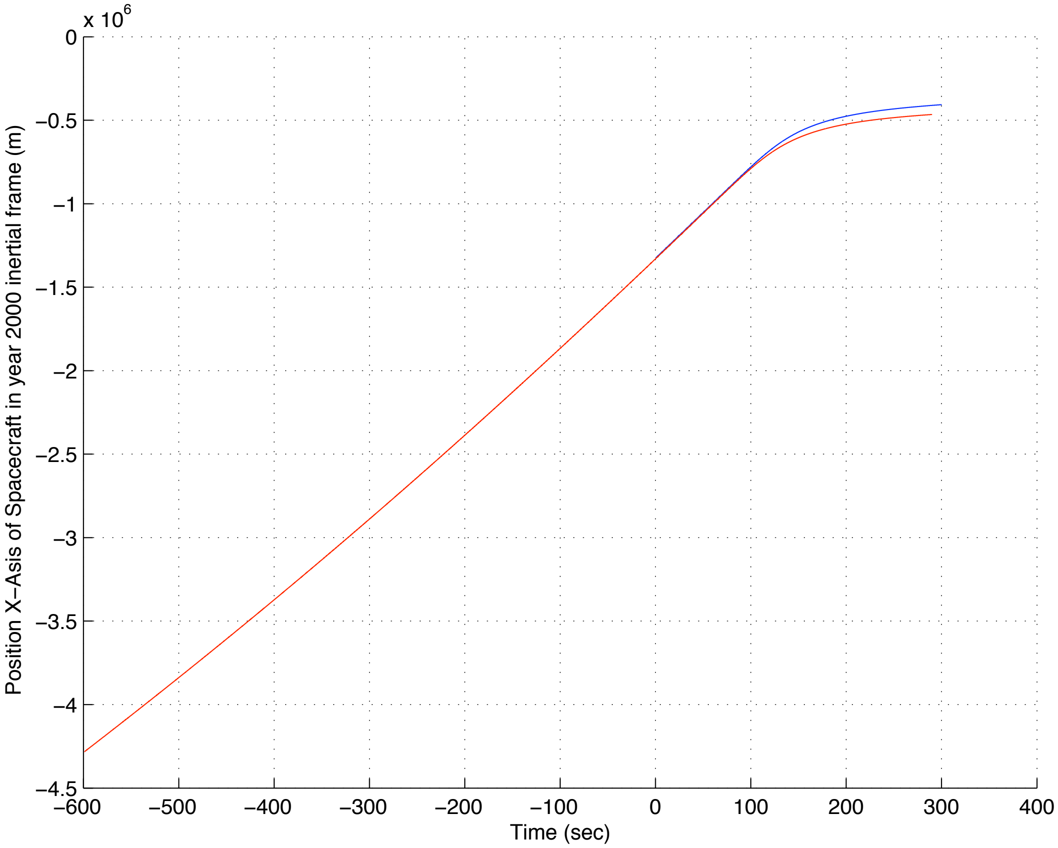

Figure A.1: Comparison of JPL MER-A Entry Trajectory and UT Simulation year 2000 x position.

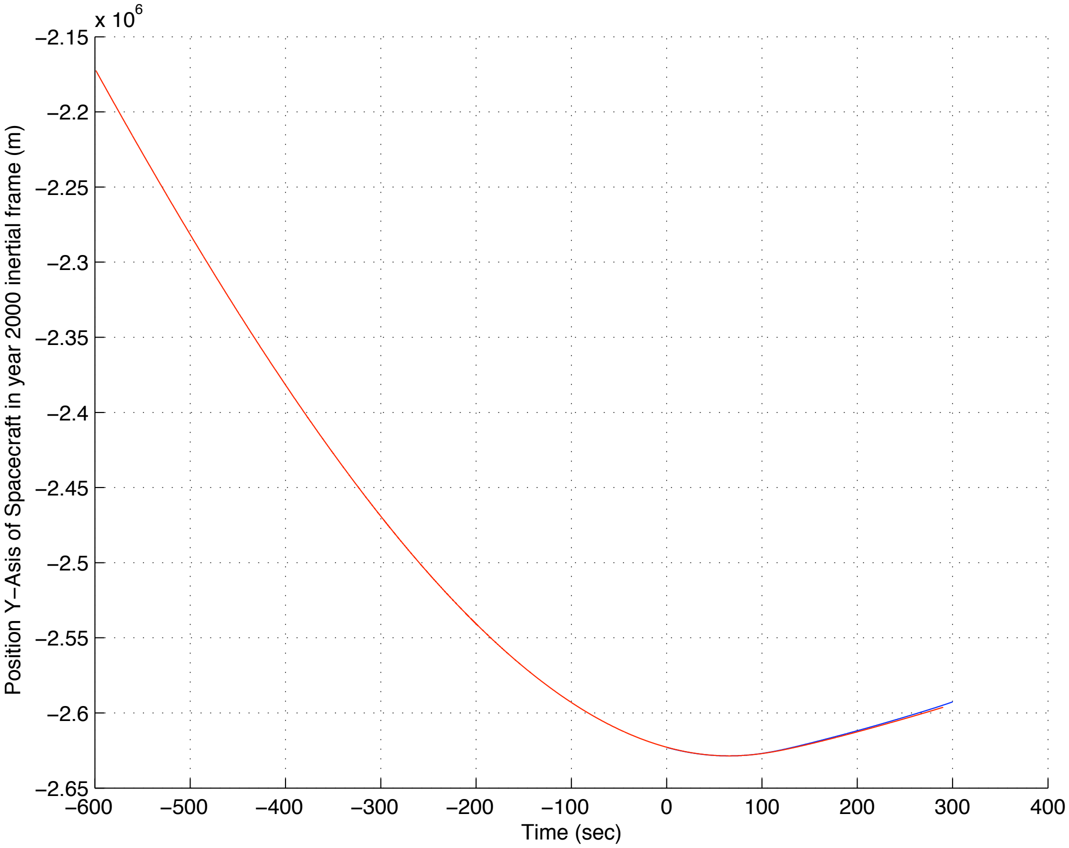

Figure A.2: Comparison of JPL MER-A Entry Trajectory and UT Simulation year 2000 y position.

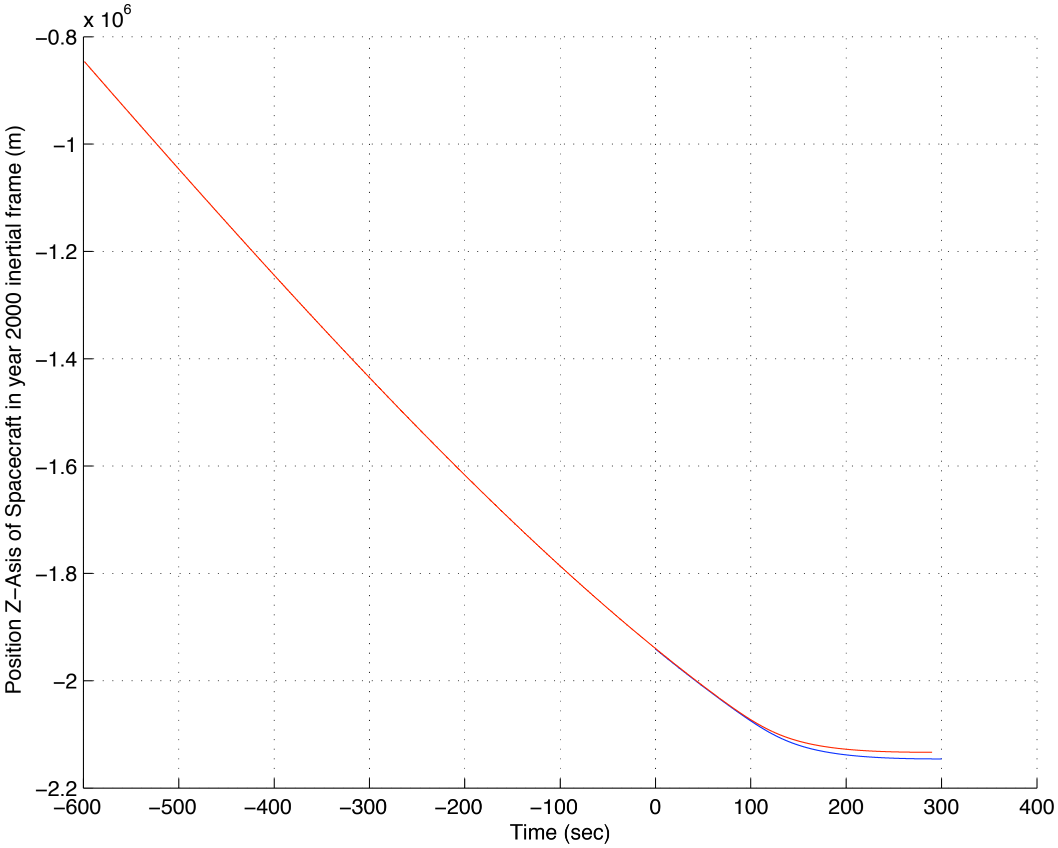

Figure A.3: Comparison of JPL MER-A Entry Trajectory and UT Simulation year 2000 z position.

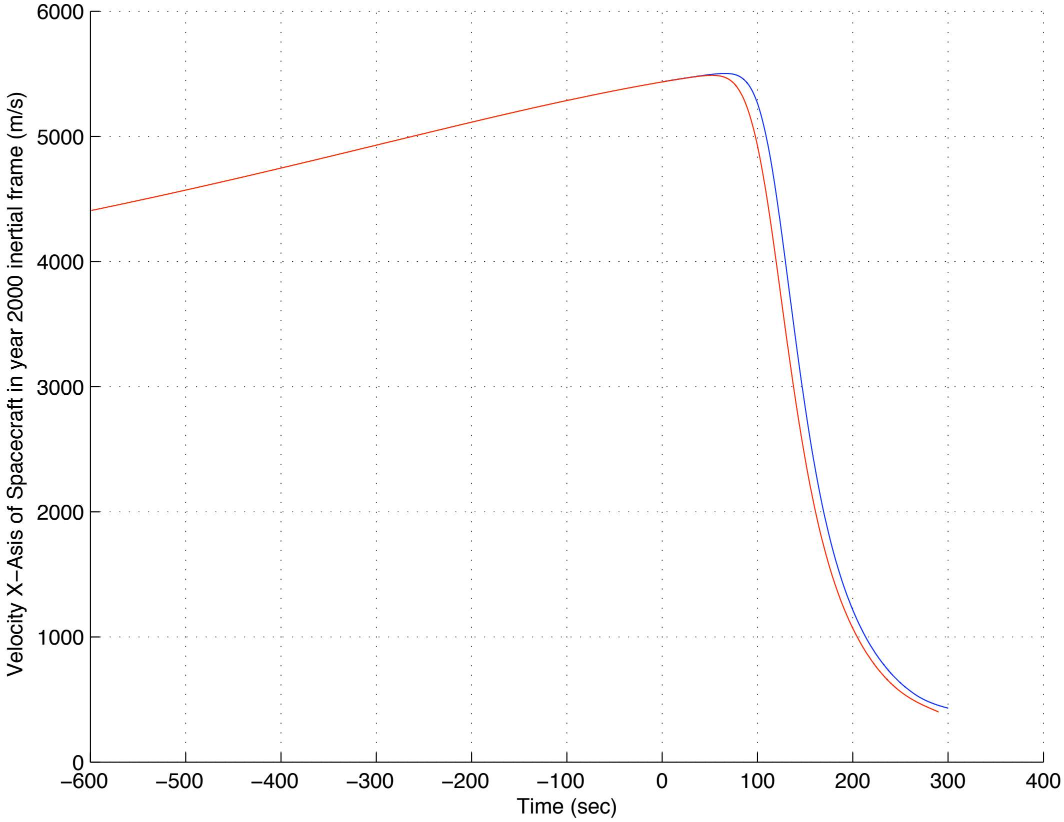

Figure A.4: Comparison of JPL MER-A Entry Trajectory and UT Simulation year 2000 x velocity.

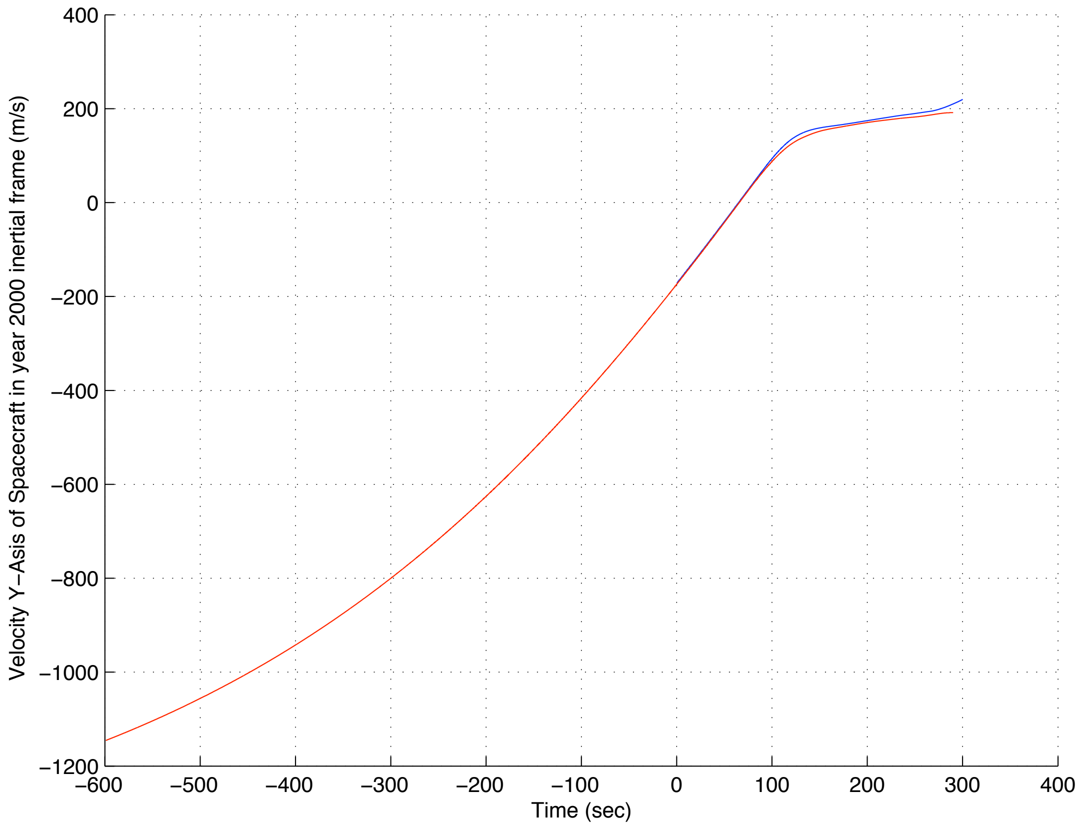

Figure A.5: Comparison of JPL MER-A Entry Trajectory and UT Simulation year 2000 y velocity.

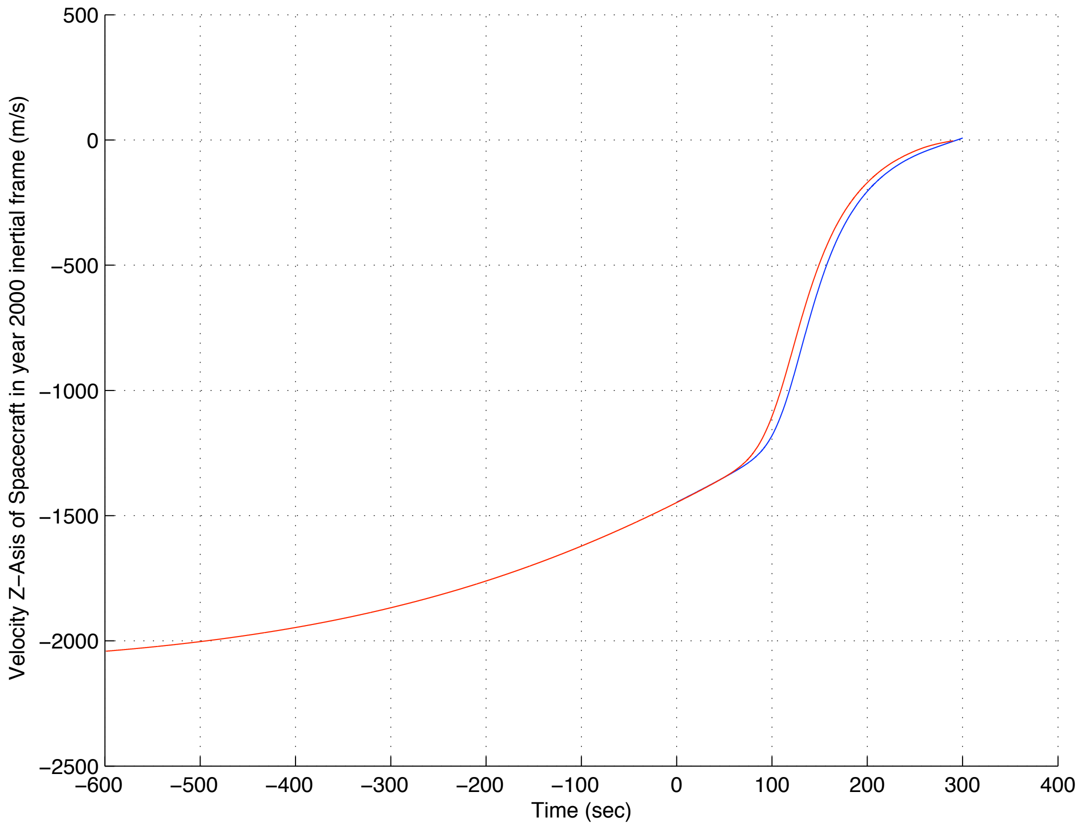

Figure A.6: Comparison of JPL MER-A Entry Trajectory and UT Simulation year 2000 z velocity.

A.1.2 UT Simulation vs. UT Reconstruction

The differences between the UT simulation and the reconstruction of the trajectory from noise free simulated IMU data are shown in Fig. (A.7) through Fig. (A.12).

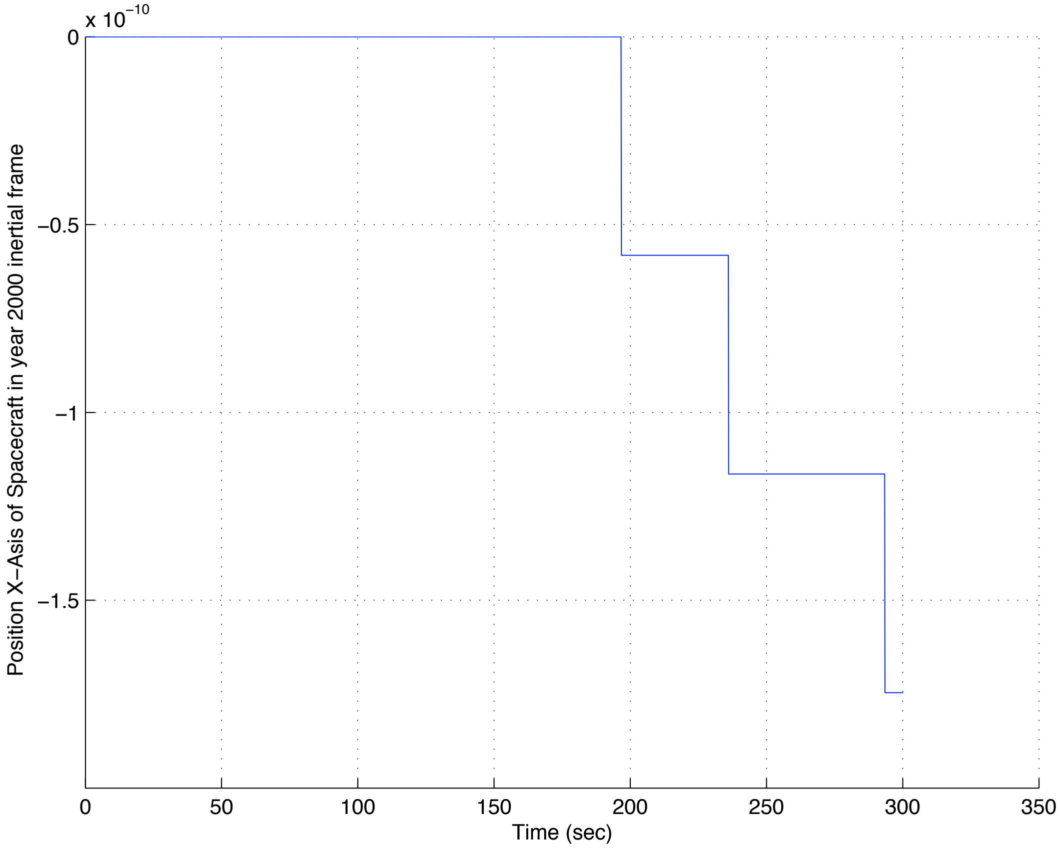

Figure A.7: Comparison of UT Simulation and UT Reconstruction year 2000 x position.

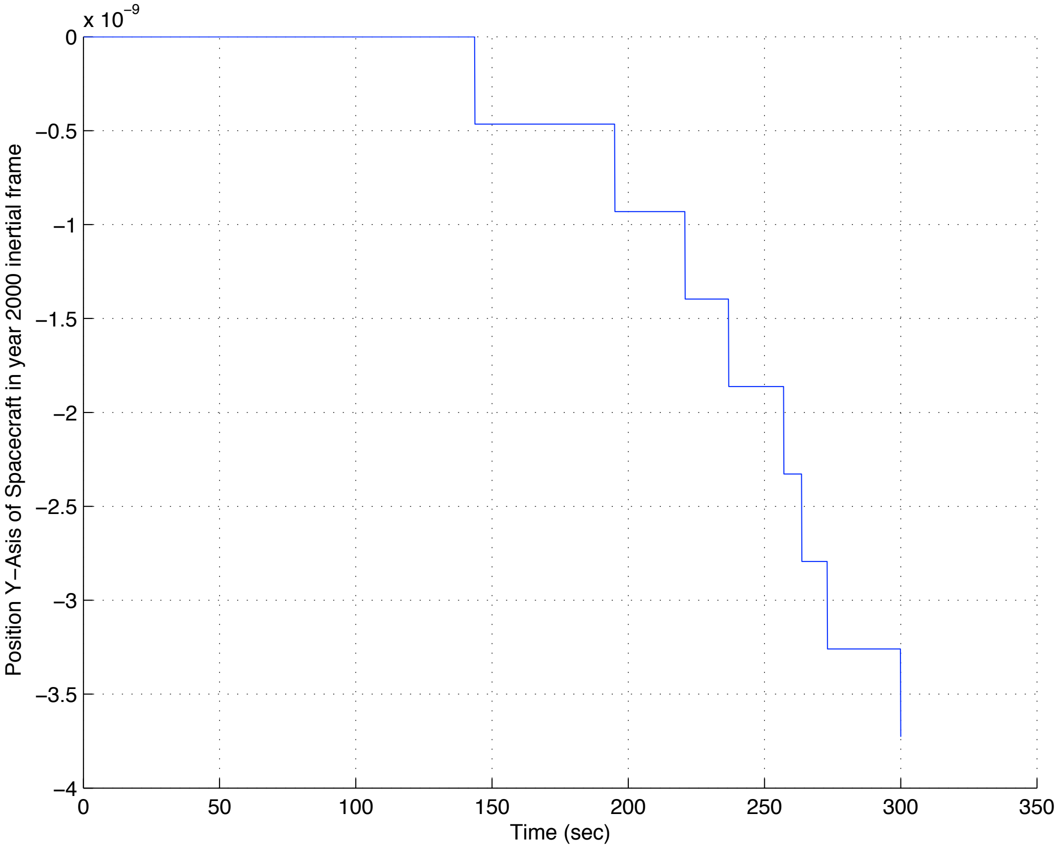

Figure A.8: Comparison of UT Simulation and UT Reconstruction year 2000 y position.

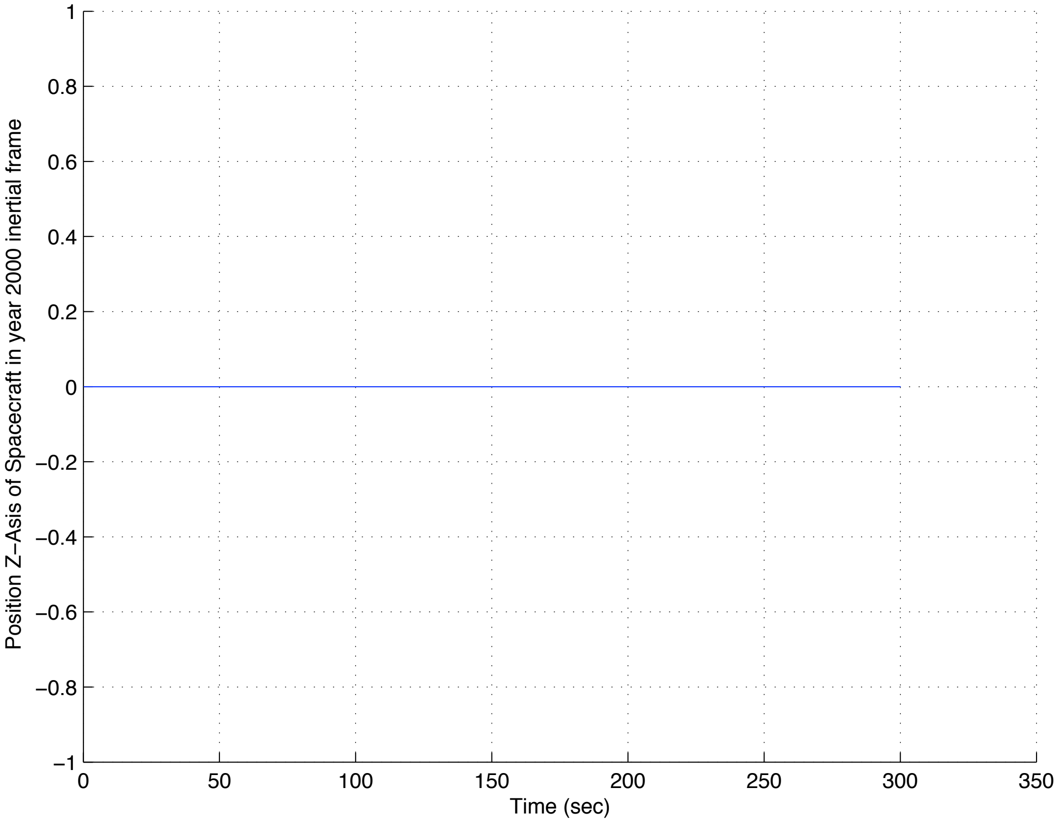

Figure A.9: Comparison of UT Simulation and UT Reconstruction year 2000 z position.

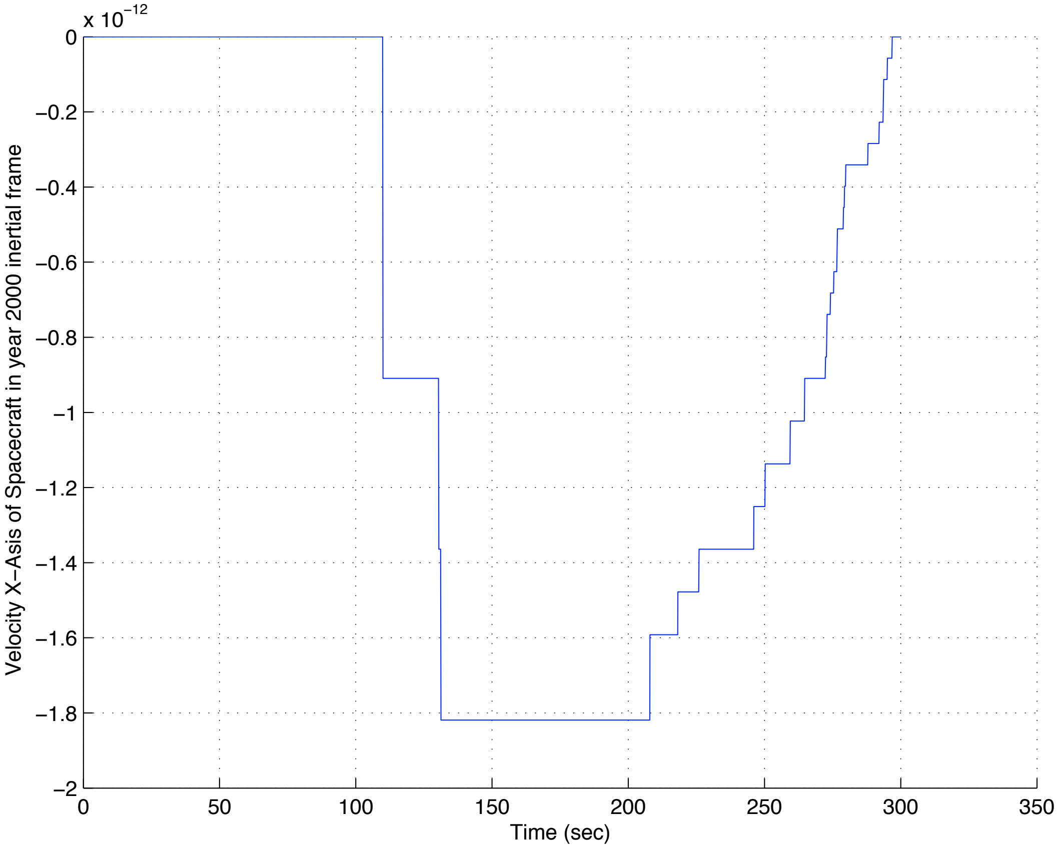

Figure A.10: Comparison of UT Simulation and UT Reconstruction year 2000 x velocity.

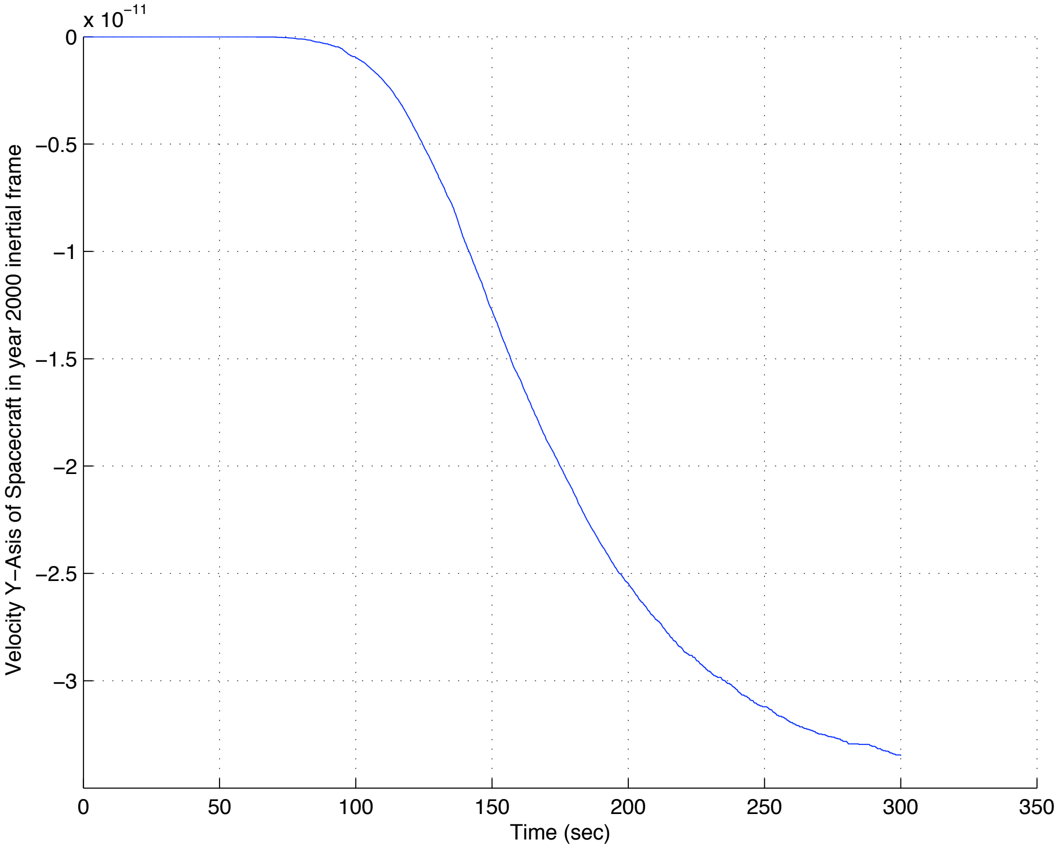

Figure A.11: Comparison of UT Simulation and UT Reconstruction year 2000 y velocity.

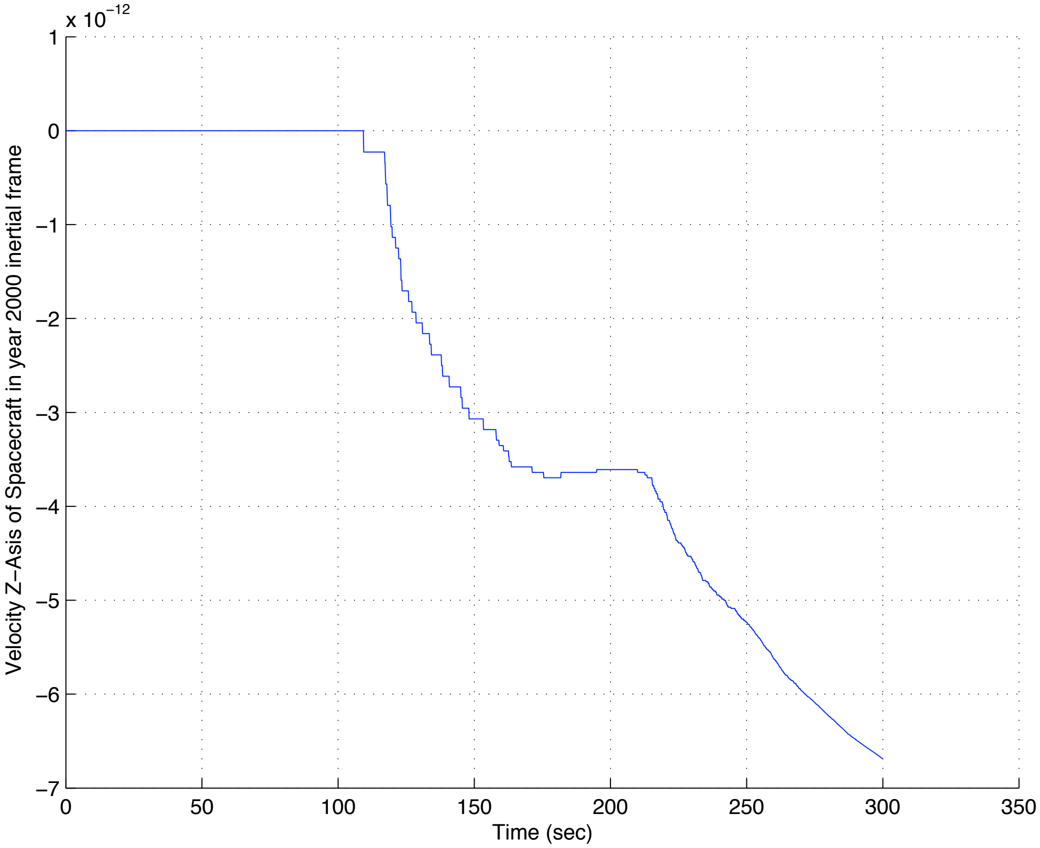

Figure A.12: Comparison of UT Simulation and UT Reconstruction year 2000 z velocity.

A.1.3 UT Simulation vs. Dead Reckoning

The differences between the UT simulation and the dead reckoned trajectory are shown in Fig. (A.13) through Fig. (A.18) for one representative dispersed initial state.

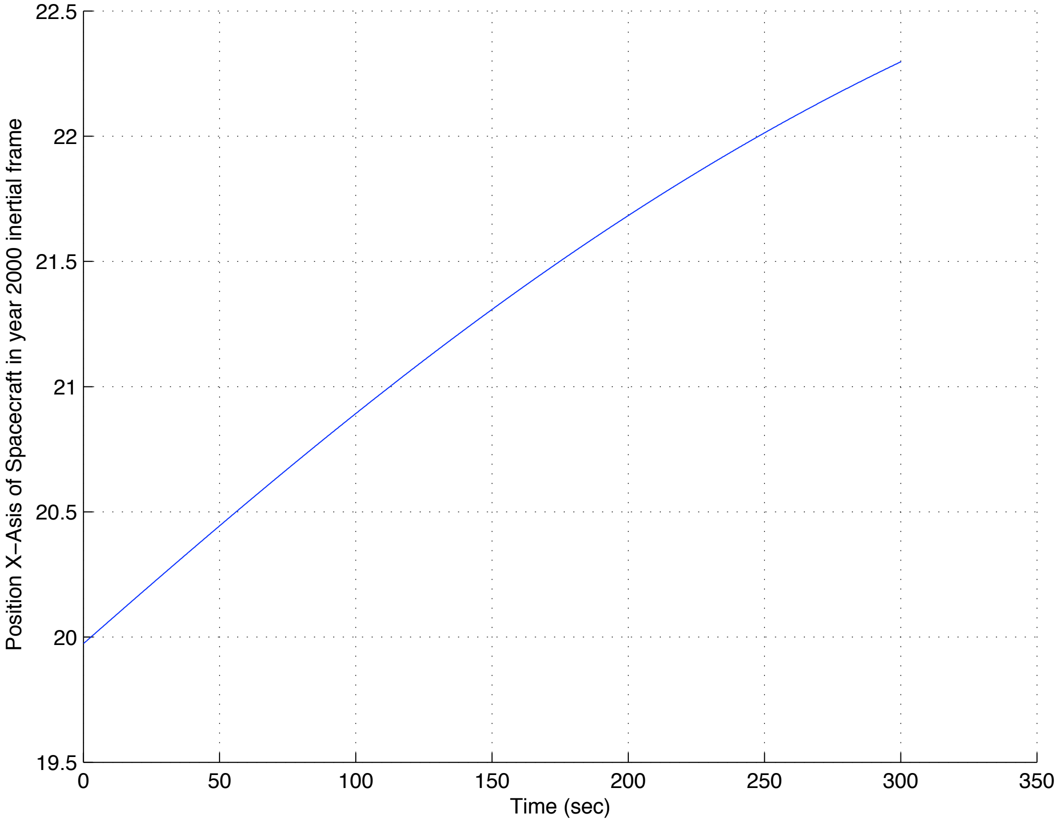

Figure A.13: Comparison of UT Simulation and Data Dead Reckoning year 2000 x position.

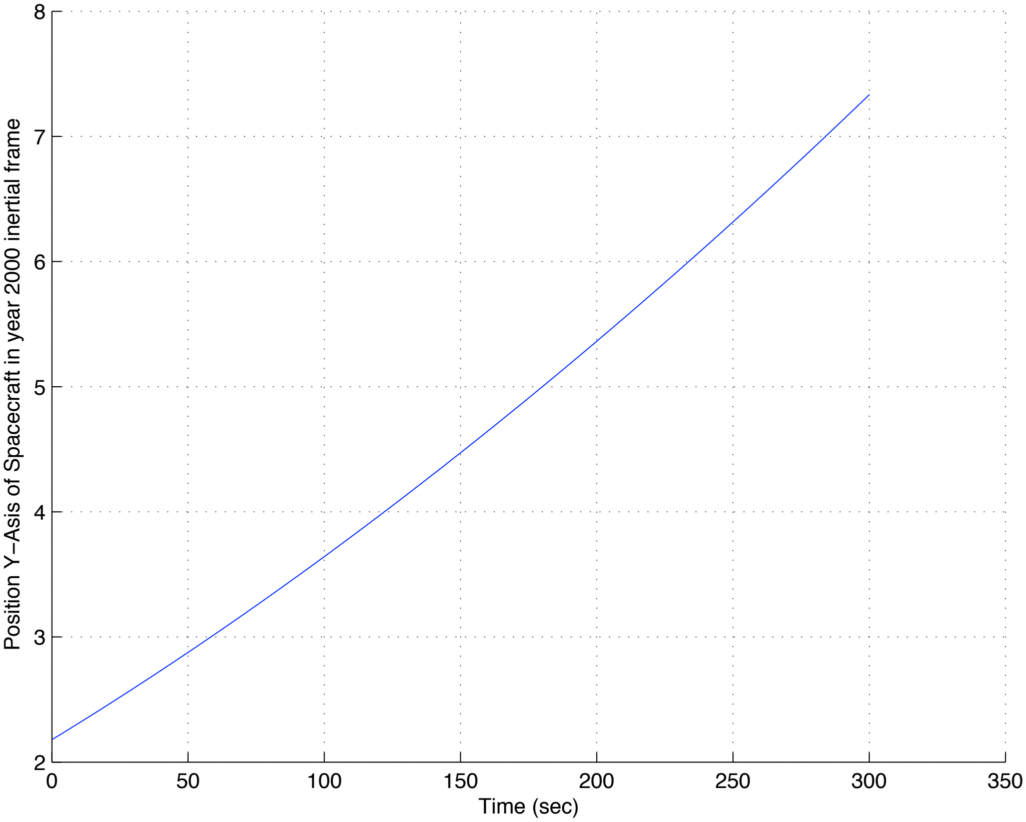

Figure A.14: Comparison of UT Simulation and Data Dead Reckoning year 2000 y position.

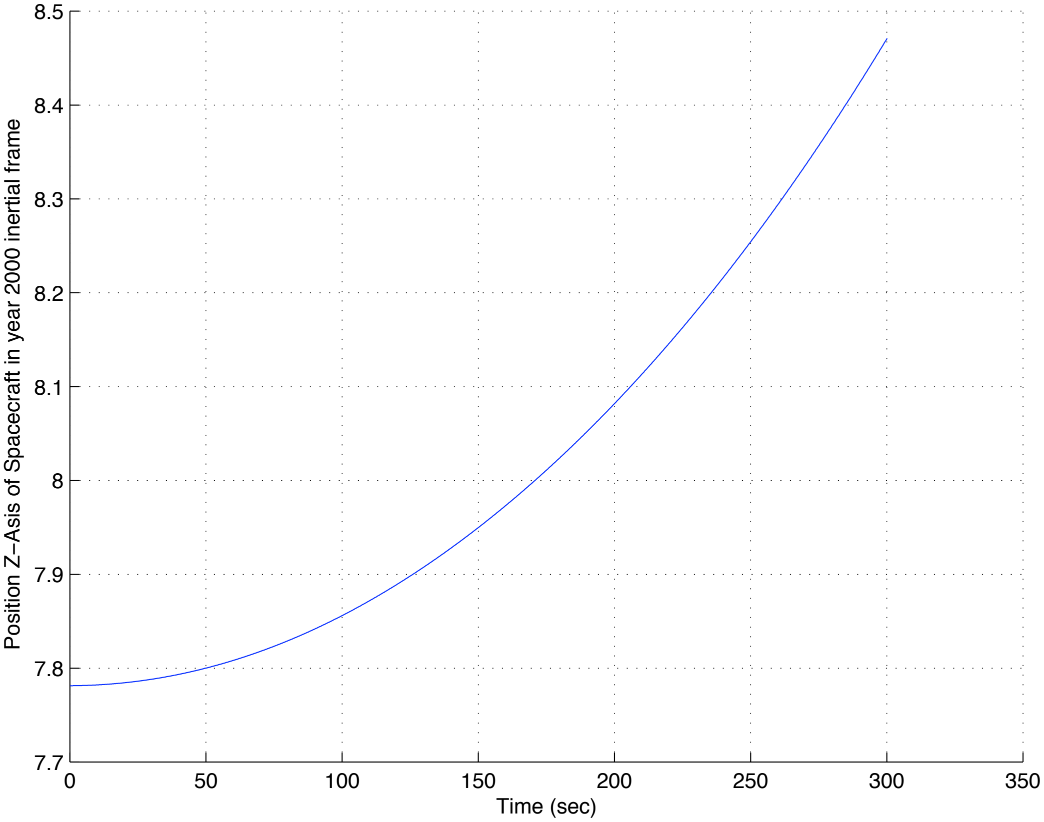

Figure A.15: Comparison of UT Simulation and Data Dead Reckoning year 2000 z position.

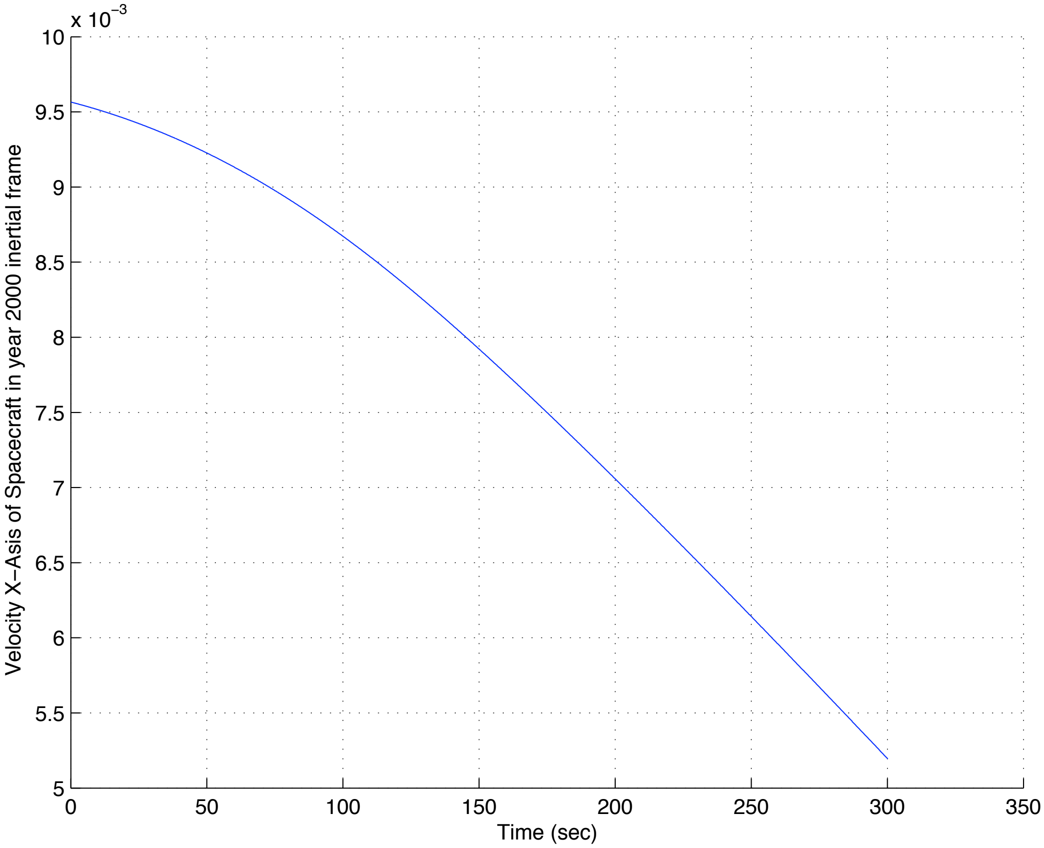

Figure A.16: Comparison of UT Simulation and Data Dead Reckoning year 2000 x velocity.

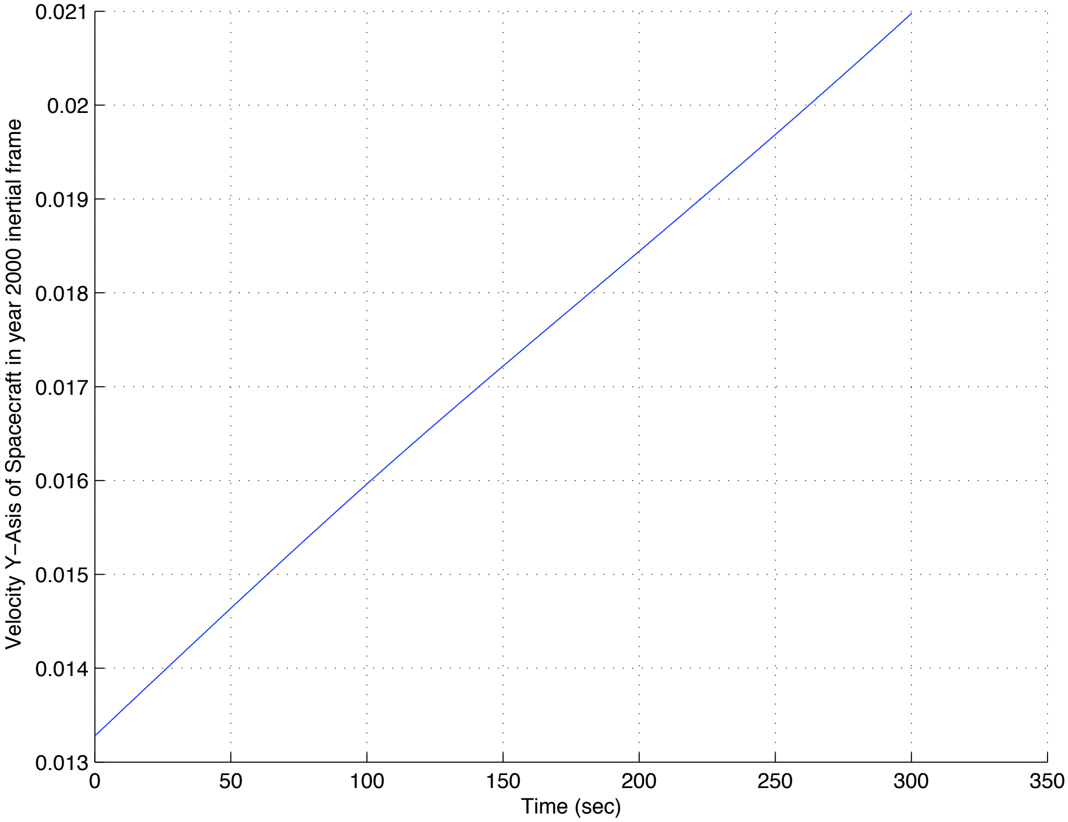

Figure A.17: Comparison of UT Simulation and Data Dead Reckoning year 2000 y velocity.

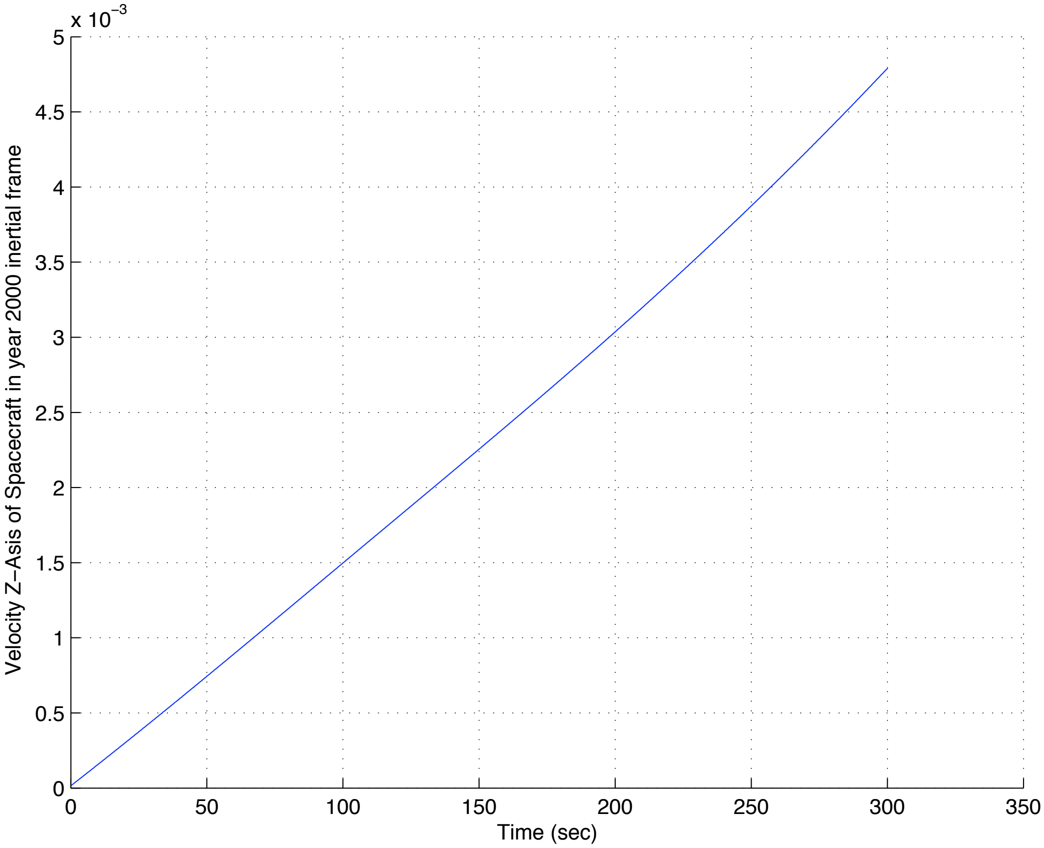

Figure A.18: Comparison of UT Simulation and Data Dead Reckoning year 2000 z velocity.

A.2 Second Scenario Trajectory Analysis

As no JPL reference was available for the second scenario only a comparison between UT simulation and dead reckoning is considered.

A.2.1 UT Simulation vs. Dead Reckoning

The differences between the UT simulation and the dead reckoned trajectory are shown in Fig. (A.19) through Fig. (A.24) for one representative dispersed initial state.

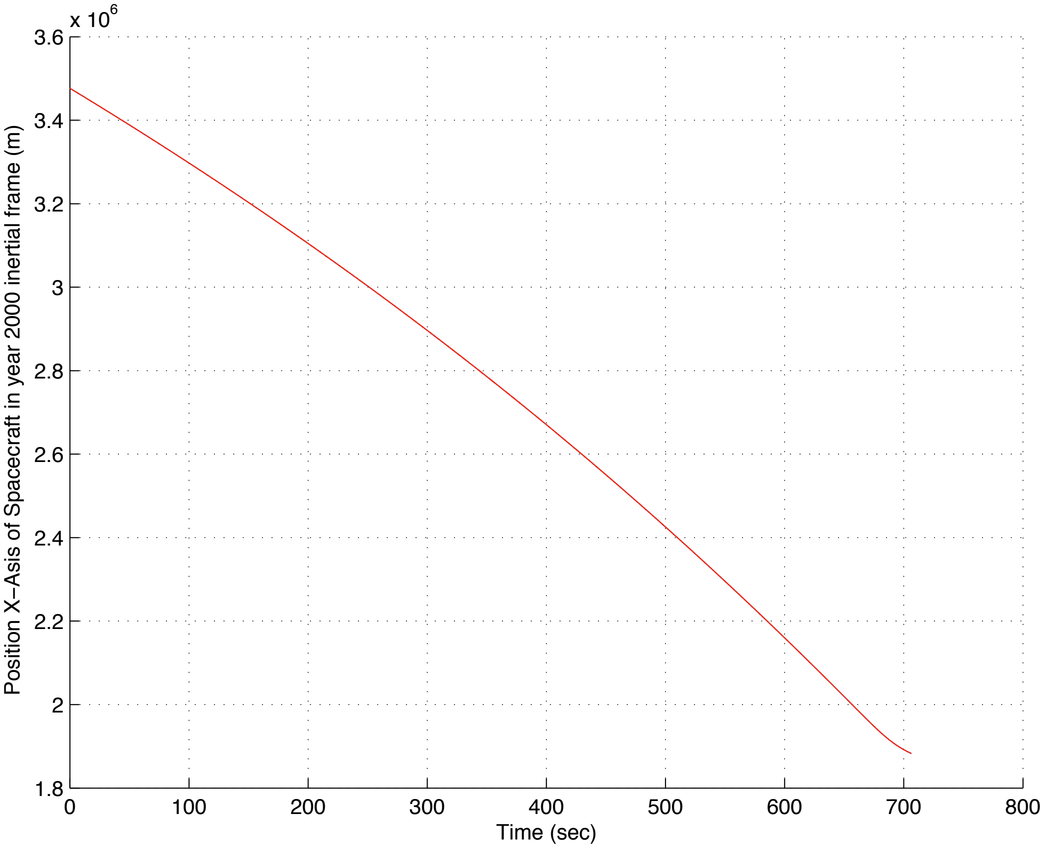

Figure A.19: Comparison of UT Second Case Simulation and Data Dead Reckoning year 2000 x position.

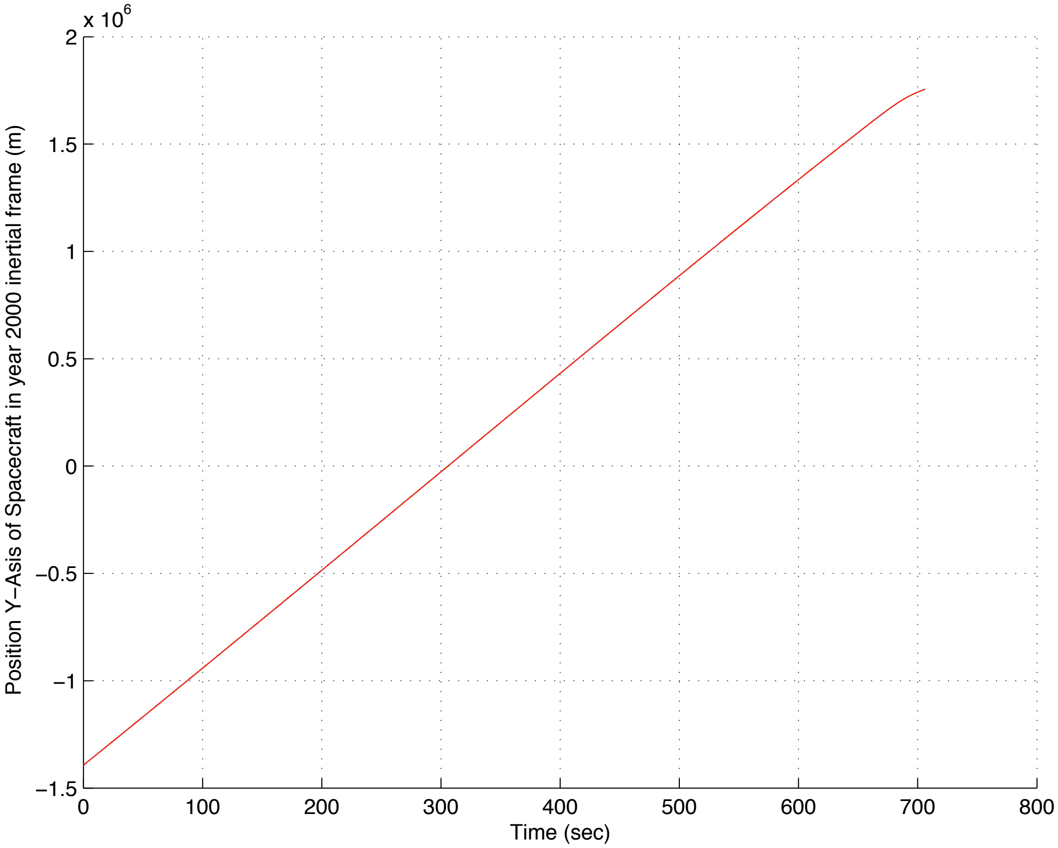

Figure A.20: Comparison of UT Second Case Simulation and Data Dead Reckoning year 2000 y position.

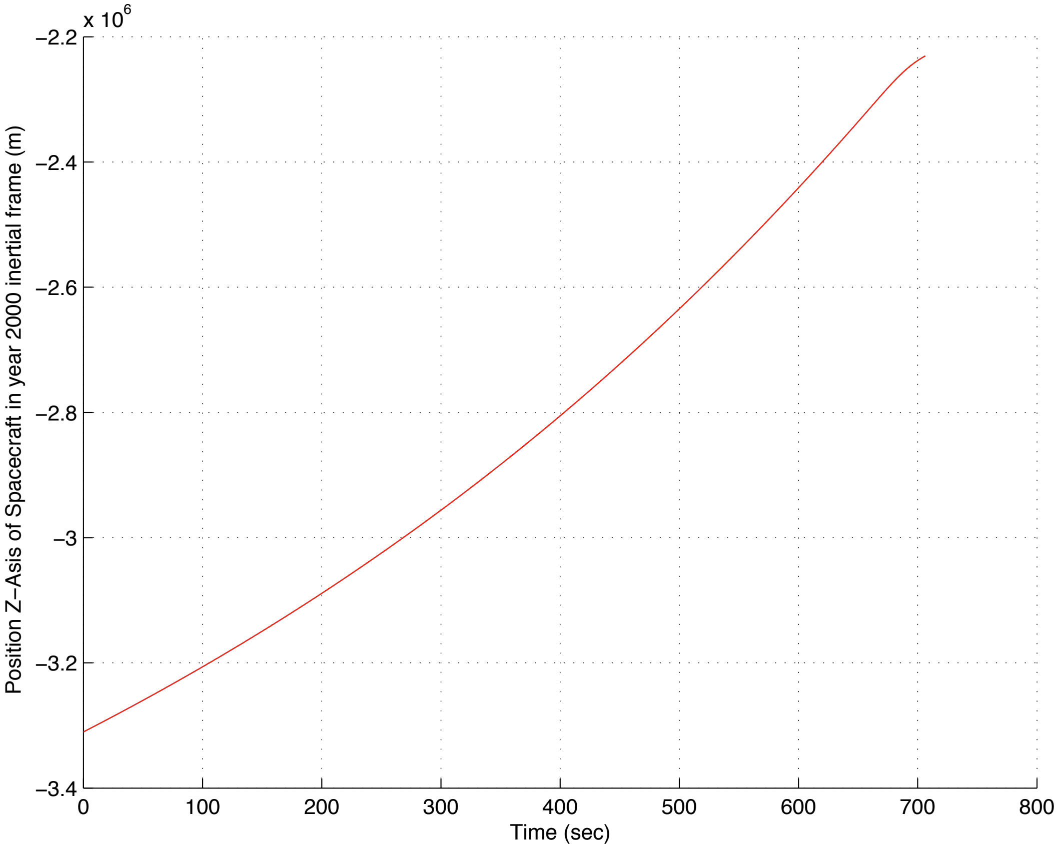

Figure A.21: Comparison of UT Second Case Simulation and Data Dead Reckoning year 2000 z position.

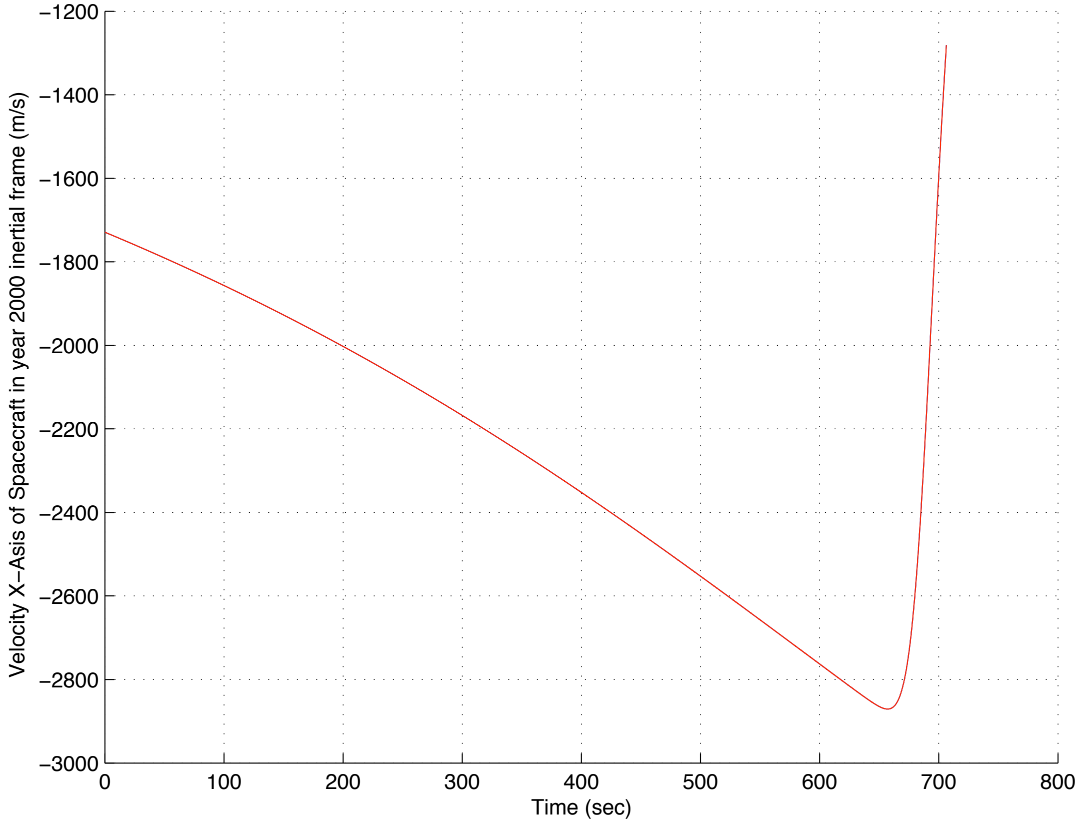

Figure A.22: Comparison of UT Second Case Simulation and Data Dead Reckoning year 2000 x velocity.

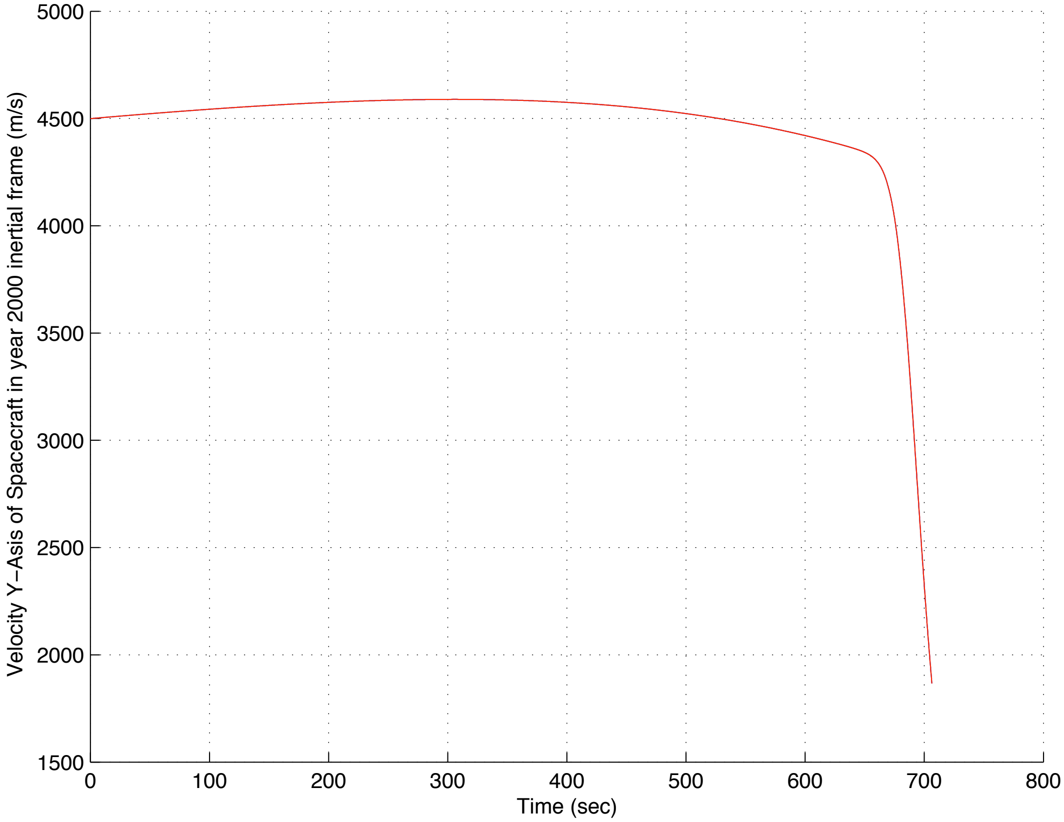

Figure A.23: Comparison of UT Second Case Simulation and Data Dead Reckoning year 2000 y velocity.

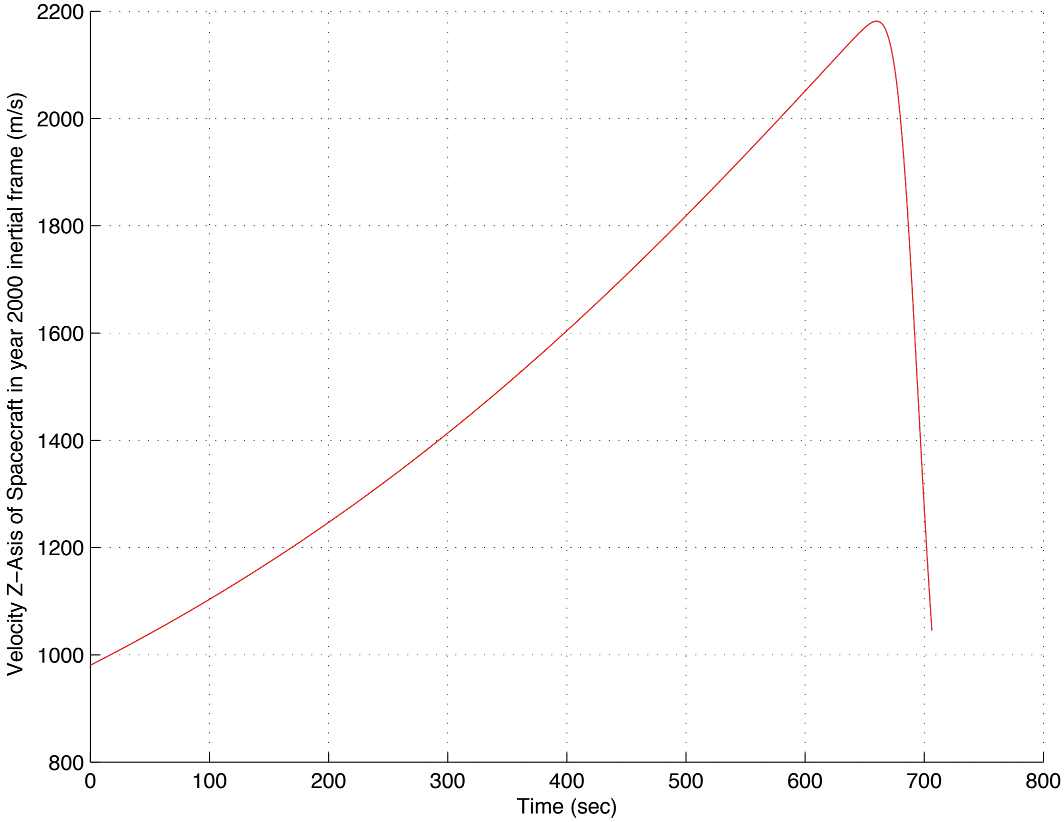

Figure A.24: Comparison of UT Second Case Simulation and Data Dead Reckoning year 2000 z velocity.

Bibliography

[1] *AFRL Web Page.* http://www.afrlhorizons.com/Briefs/Mar02/OSR0106.html.

[2] *Guidance systems Web Page.* http://en.wikipedia.org/wiki/Guidance_system.

[3] *Mars Exploration Rover Web Page.* http://mars.jpl.nasa.gov/mer/overview/.

[4] *Mars Science Laboratory Web Page.* http://mars.jpl.nasa.gov/msl/overview/.

[5] *Navigation Web Page.* http://en.wikipedia.org/wiki/Navigation.

[6] Guidance and navigation for entry vehicles. Technical Report NASA SP-8015, 1968.

[7] Richard H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition.* AIAA Education Series, 1999.

[8] R. C. Blanchard and G. D. Walberg. Determination of the hypersonic / rarefield-flow drag coefficient of the viking lander capsule 1 aeroshell from flight data. Technical Report NASA Technical Paper TP-1793, 1980.

[9] T. P. Crain. Adaptive interplanetary orbit determination. Technical report, PhD Dissertation, The University of Texas at Austin, 2000.

[10] John L. Crassidis and John L. Junkins. *Optimal Estimation of Dynamic Systems.* Chapman, Hall, 2004.

[11] P. N. Desai and P. C. Knocke. Mars exploration rovers entry, descent, and landing trajectory analysis. Technical Report 2003-2125, AIAA, Monterey, CA, 2003.

[12] P. N. Desai and P. C. Knocke. Mars exploration rovers entry, descent, and landing trajectory analysis. Technical Report 2004-5092, AIAA, Washington, DC, 2004.

[13] Raynor L. Duncombe. Determination of time. Technical report, The Universtiy of Texas at Austin.

[14] A. Gelb. *Applied Optimal Estimation*. The M.I.T. Press, Cambridge, MA, 1974.

[15] Martin C. Heyne and Robert H. Bishop. Spacecraft entry navigation using sigma point kalman filtering. Technical report, Institute of Navigation, Coronado, CA, April 2006.

[16] Martin C. Heyne and Robert H. Bishop. Integrated spacecraft entry navigation on the electra programmable radio. Technical report, Institute of Navigation, San Diego, CA, January 2007.

[17] S. J. Julier. The scaled unscented transformation. Technical report, American Control Conference, 2002.

[18] Oliver Montenbruck and Everhard Gill. *Satellite Orbits*. Springer, 2nd edition, 2001.

[19] R. Morth. Apollo guidance and navigation: Reentry guidance for apollo. Technical Report Technical Paper R-532, MIT Instrumentation Laboratory, 1966.

[20] Charles Murray and Catherine Bly Cox. *Apollo - The Race to the Moon.* Simon, Schuster, 1989.

[21] NASA/JPL. *Electra Programmable Radio User Manual.*

[22] R. A. Micheltree P. N. Desai and F. McNeil Cheatwoot. Entry dispersion analysis for the stardust comet sample return capsule. Technical Report Vol. 36, No. 3, pp. 463-469, AIAA Journal of Spacecraft and Rockets, 1999.

[23] G. Peterson and R. H. Bishop. Entry trajectory dispersions due to uncertainties in the mars atmosphere. Technical Report 98-147, AAS/AIAA, Monterey, CA, 1998.

[24] O. Dubois-Matra R. H. Bishop and T. Ely. Robust entry navigation using hierarchical filter architectures regulated with gating networks. Technical report, 16th International Symposium on Space Flight Dynamics, 2001.

[25] R. B. Roncoli and J. M. Ludwinski. Mission design overview for the mars exploration rover mission. Technical Report 2002-4823, AIAA, Monterey, CA, 2002.

[26] Dyron D. Tapley and Bob E. Schutz. *Statistical Orbit Determination.* Academic Press, 2000.

[27] David A. Vallado. *Fundamentals of Astrodynamics and Applications*. McGraw Hill, 1997.

[28] E. A. Wan and R. van der Mere. *Kalman Filtering and Neural Networks*. Wiley, 2001.

Vita

Martin Cornelius Heyne was born in Hünfeld, Germany on January 22^{nd}, 1973 as the son of Ulrich Heyne and Gerhild Erna Heyne. After attending Stadium High School in Tacoma, WA, he graduated from Ferdinand-Braun-Schule in Fulda, Germany, with highest honors and began his studies of aerospace engineering at the Universität Stuttgart, Germany, from which he graduated as Diplom-Ingenieur. He worked at PT. Industri Pesawat Terbang Nusantara in Bandung, Indonesia, and did reseach on missile guidance at the Institute of Flight Mechanics at the University of Stuttgart. In 1999 he worked at NASA's Johnson Space Center in Houston, TX, where he wrote his Master's Thesis. He entered the Ph.D. program at the University of Texas at Austin in 2000 working under Dr. Robert H. Bishop. His research interest is spacecraft navigation and the computational implementation of navigation algorithms for the Entry, Descent and Landing problem.

Permanent address: Leimbachshöfer Straße 4
36088 Hünfeld
Germany

This dissertation was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.